

Anwendung stabilisierter Newton-artiger Verfahren zum Lösen der Navier-Stokes Gleichung

Masterarbeit
von
cand. aer. Shunsaku Kobayashi

05-08-2018

durchgeführt am
Institut für Aerodynamik und Gasdynamik
der Universität Stuttgart
und am Deutschen Zentrum für Luft- und Raumfahrt
Institut für Aerodynamik und Strömungstechnik
Center for Computer Applications in AeroSpace Science and Engineering.

Stuttgart, im August 2018

Aufgabenstellung

Thema für Masterarbeit für Herrn Shunsaku Kobayashi

angefertigt am Deutschen Zentrum für Luft- und Raumfahrt
Institut für Aerodynamik und Strömungstechnik
Center for Computer Applications in AeroSpace Science and Engineering

Thema:

Anwendung stabilisierter Newton-artiger Verfahren zum Lösen der Navier-Stokes Gleichung

Zum Lösen der Reynolds gemittelten Navier-Stokes Gleichungen haben sich implizite Verfahren, die als Glätter in einem Mehrgitterverfahren genutzt werden, als vorteilhaft erwiesen. Die Konstruktion geeigneter Glätter hängt von vielen Parametern und Festlegungen ab. Allgemein können die Glätter als mehrstufige Runge-Kutta Verfahren

$$\begin{aligned}\mathbf{W}^{(0)} &:= \mathbf{W}^{\mathbf{T}_n} \\ \mathbf{W}^{(1)} &= \mathbf{W}^{(0)} - \alpha_{21}\Delta t \left[\frac{dg_1(k^{(0)})}{dk} \right]^{-1} \mathbf{M}^{-1} \mathbf{R}(\mathbf{W}^{(0)}) \\ &\vdots \\ \mathbf{W}^{(s)} &:= \mathbf{W}^{(0)} - \alpha_{s+1,s}\Delta t \left[\frac{dg_s(k^{(0)})}{dk} \right]^{-1} \mathbf{M}^{-1} \mathbf{R}(\mathbf{W}^{(s-1)}) \\ \mathbf{W}^{\mathbf{T}_{n+1}} &= \mathbf{W}^{(s)}.\end{aligned}\tag{0.1}$$

ausgedrückt werden. Hierbei bezeichnet \mathbf{R} das Residuum, welches sich aus der Diskretisierung ergibt, \mathbf{M} ist die Massenmatrix und

$$g_j(k) := k_j + \mathbf{M}^{-1} \mathbf{R}(\mathbf{W}^{\mathbf{T}_n} + \alpha_{j,j-1}\Delta t k_{j-1} + \alpha_{jj}\Delta t k)\tag{0.2}$$

ist eine nichtlineare Gleichung, die in jedem Schritt approximativ gelöst werden muss. Die Ableitung ist gegeben durch

$$\frac{dg_j(k)}{dk_j} = \mathbf{I} + \alpha_{jj}\Delta t \mathbf{M}^{-1} \frac{d\mathbf{R}}{d\mathbf{W}}(\mathbf{W}^{\mathbf{T}_n} + \alpha_{j,j-1}\Delta t k_{j-1} + \alpha_{jj}\Delta t k).\tag{0.3}$$

Dieses bedeutet, dass für jede Stufe des Runge-Kutta Verfahrens ein lineares Gleichungssystem der Form

$$\left((\Delta t)^{-1} \mathbf{M} + \alpha_{jj} \frac{d\mathbf{R}}{d\mathbf{W}} \right) \mathbf{h}_j = \alpha_{j+1,j} \mathbf{R}(\mathbf{W}^{(j-1)}). \quad (0.4)$$

approximativ gelöst werden muss, mit einer geeigneten Zeitschrittwahl Δt .

Statt der direkten Anwendung eines Newton-ähnlichen Verfahrens, kann das Lösen auch durch einen Levenberg-Marquardt Algorithmus erfolgen. D.h., es soll das Minimierungsproblem

$$\left\| \frac{d\mathbf{R}}{d\mathbf{W}}(\mathbf{W}^n) \mathbf{h}_n + \mathbf{R}(\mathbf{W}^n) \right\|^2 + \gamma_n \|\mathbf{h}_n\|^2 = \min_{h_n}! \quad (0.5a)$$

$$\mathbf{W}^{n+1} = \mathbf{W}^n + \mathbf{h}_n. \quad (0.5b)$$

gelöst werden. Hierbei bezeichnet $\{\gamma_n\}_{n \in \mathbb{N}_0}$ eine Folge von Parametern $\gamma_n \geq 0$. Unter Kenntnis, dass das Minimierungsproblem (0.5a) äquivalent ist zu

$$\left\| \begin{pmatrix} \frac{d\mathbf{R}}{d\mathbf{W}}(\mathbf{W}^n) \\ \sqrt{\gamma_n} \mathbf{I} \end{pmatrix} \mathbf{h}_n + \begin{pmatrix} \mathbf{R}(\mathbf{W}^n) \\ 0 \end{pmatrix} \right\| = \min_{h_n}!. \quad (0.6)$$

kann das eindeutig bestimmte Minimum durch Lösen der Normalengleichungen

$$\left(\gamma_n \mathbf{I} + \left(\frac{d\mathbf{R}}{d\mathbf{W}} \right)^T \frac{d\mathbf{R}}{d\mathbf{W}} \right) \mathbf{h}_n = - \left(\frac{d\mathbf{R}}{d\mathbf{W}} \right)^T \mathbf{R}(\mathbf{W}^n) \quad (0.7)$$

erfolgen. Dieses kann z.B. durch eine Anwendung des Verfahren der konjugierten Gradienten erfolgen. Der Regularisierungsparameter γ_n übernimmt hierbei die Rolle der Zeitschrittwahl.

Aufbauend auf bestehende Implementierungen für Algorithmen (0.1) und (0.4) besteht die Aufgabe darin, einen Levenberg-Marquardt Algorithmus in einen bestehenden Lösungsalgorithmus zu implementieren und zu integrieren. Die Funktionsfähigkeit wird an grundlegenden 2D-Konfigurationen sowie an 3-D Testfällen demonstriert. Darüberhinaus sollen verschiedene Wahlen von Regularisierungsparametern untersucht werden und mit der üblichen Zeitschrittwahl verglichen werden. Die Ergebnisse werden im Rahmen einer Masterarbeit dokumentiert und zusammengefasst.

Ausgabedatum:

Abgabedatum:

Der interne Betreuer: Prof. Dr. rer. nat. **Claus-Dieter Munz**

Der externe Betreuer: Dr. habil. **Stefan Langer**

Der Studierenden: **Shunsaku Kobayashi**

Übersicht

Im Rahmen eines impliziten Runge-Kutta-Verfahrens, das als Glätter für Mehrgitterverfahren zum Lösen der Reynolds gemittelten Navier-Stokes Gleichung (RANS) verwendet wird, wird ein lineares Gleichungssystem in jeder Stufe approximativ gelöst. In der aktuellen Forschung wird hierfür direkt ein Newton Verfahren angewendet. Die schnelle Konvergenz durch Newton-Verfahren ist aber nur in der Nähe der richtigen Lösung und die Berechnung kann nur lokal konvergiert werden. Außerdem wird das Newton-Verfahren nur verwendet, wenn man die Bestätigung guter Konvergenzen im Testfall hat. Um diese Lösungsmethode zu verbessern, kann die Lösung des linearen Systems durch den Levenberg-Marquardt Algorithmus statt der direkten Anwendung eines Newton-Verfahrens erfolgen. In dieser Masterthesis wird der Levenberg-Marquardt Algorithmus für sowohl ohne Parallelisierung als auch Parallelisierung implementiert.

In numerischen Beispielen werden die beiden Methoden als 2D-Testfall für das NACA0012 Profil mit Eulerer- und laminarer Strömungen getestet. Die Konvergenz der Lösung mit dem Levenberg-Marquardt Algorithmus erfordert generell hohe CFL-Zahlen, die als ein gewichtet Faktor von Regularisierungsparameter γ eine Rolle spielt. Die Ergebnisse im 2D Testfalls für ohne Parallelisierung bestätigen, dass der Levenberg-Marquardt Algorithmus anwendbar für Testfälle ohne Parallelisierung ist und die Lösung mit 10 bis 1000 CFL-Zahl konvergiert wird. Außerdem werden der Widerstandsbeiwert und der Auftriebsbeiwert durch die beiden Methoden zu gleiche Wert konvergiert. Im 3D-Testfall wird die Lösungsmethode durch direkte Anwendung eines Newton-Verfahrens für turbulente Strömung getestet, die auf dem negativen Spalart-Allmaras Turbulenzmodell basiert.

Inhaltsverzeichnis

Aufgabenstellung	ii
Übersicht	iv
Inhaltsverzeichnis	v
Nomenklatur	vi
1 Einleitung	1
2 Versuchsanordnung bzw. verwendete Software	4
2.1 Grundlegende Gleichungen	4
2.1.1 Reynolds-gemittelte Navier-Stokes-Gleichungen	4
2.1.2 Turbulenzmodelle	5
2.2 Diskretisierung	7
3 Numerische Lösungsmethode	13
3.1 Nichtlineares Mehrgitterverfahren	14
3.1.1 Grobgittergleichung	17
3.2 Implizites Runge Kutta Verfahren	19
3.3 Levenberg-Marquardt Algorithmus	23
3.4 Lösungsmethode für das linear Gleichungssystem	25
4 Numerische Beispiele	27
4.1 2D-Testfälle	28
4.1.1 Testfall(a):NACA0012, Euler-Strömung	29
4.1.2 Testfall(b):NACA0012, Laminare Strömung	35
4.2 3D-Testfälle	43
4.2.1 Testfall(c):Hemisphere Zylinder	43
5 Zusammenfassung	48
Literaturverzeichnis	49
Anhang	51

Nomenklatur

A	[-]	Koeffizientenmatrix der implizites Runge-Kutta Verfahren
\mathbf{A}^{Roe}	[-]	Die Roe Matrix
AoA	[°]	Anstellwinkel
a	[m/s]	Schallgeschwindigkeit
\mathbf{b}	[-]	Koeffizientevektor der implizites Runge-Kutta Verfahren
CFL	[-]	Die Courant-Friedrichs-Lewy-Zahl
$C_j^{(k)}$	[-]	i-te Grogitterzell auf der Gitterstufe k
c	[-]	Koeffizientevektor der implizites Runge-Kutta Verfahren
c_{b1}	[-]	ein Konstant für Spalart-Allmaras Turbulenzmodell
c_{b2}	[-]	ein Konstant für Spalart-Allmaras Turbulenzmodell
c_d	[-]	Widerstandbeiwert
$c_{d,p}$	[-]	Drückwiderstand
$c_{d,v}$	[-]	Reibungswiderstand
c_f	[-]	Reibungsbeiwert
c_{n1}	[-]	ein Konstant für Spalart-Allmaras Turbulenzmodell
c_{t3}	[-]	ein Konstant für Spalart-Allmaras Turbulenzmodell
c_{t4}	[-]	ein Konstant für Spalart-Allmaras Turbulenzmodell
c_l	[-]	Auftriebbeiwert
c_p	[-]	Druckbeiwert
c_{v1}	[-]	ein Konstant für Spalart-Allmaras Turbulenzmodell
c_{v2}	[-]	ein Konstant für Spalart-Allmaras Turbulenzmodell
c_{v3}	[-]	ein Konstant für Spalart-Allmaras Turbulenzmodell
c_{w1}	[-]	ein Konstant für Spalart-Allmaras Turbulenzmodell
c_{w2}	[-]	ein Konstant für Spalart-Allmaras Turbulenzmodell
c_{w3}	[-]	ein Konstant für Spalart-Allmaras Turbulenzmodell
\mathbf{D}	[-]	Künstliche Viskosität
D	[-]	Diagonalmatrix
De	[-]	Destruktion
Di	[-]	Diffusion
d	[-]	Freiheitsgrad
d	[m]	Wandabstand
E	[Nm]	spezifische totalen Energie
\mathbf{f}	[-]	rechtseitige Vektor der nichtlineare Gleichung
\mathbf{f}_c	[-]	Der konvektive Fluss
$\mathbf{f}_{c,turb}$	[-]	Der turbulenten konvektiven Fluss
f_{t2}	[-]	laminarer Unterdrückungsterm für Spalart-Allmaras Turbulenzmodell

f_{v_1}	[-]	Funktion der Wirbelviskositätskorrelation
f_{v_2}	[-]	Funktion der Wirbelviskositätskorrelation
\mathbf{f}_v	[-]	Der viskose Fluss
$\mathbf{f}_{v,turb}$	[-]	Der turbulenten viskose Fluss
G	[-]	Punkt im Dualgitter
H	$[kgm^2/s^2]$	totale Enthalpie
\mathbf{h}	[-]	Die Lösungsvektor der linear Gleichungssystem
h	[-]	Längeneinheit
\mathbf{I}	[-]	Interpolationsoperator
k	[-]	Nummer der Gitter
k	[-]	Die Stufekoeffizient der implizites Runge-Kutta Verfahren
\mathbf{L}	[-]	untere Dreiecksmatrix
\mathbf{M}	[-]	Die Massenmatrix
M_n	[-]	Das n-te stüfigen Gitter
Ma	[-]	Mach-Zahl
N	[-]	Anzahl der Gitterpunktes
$N(i)$	[-]	Die Nachbarpunkte von I
$\bar{\mathbf{n}}$	[-]	nomalen Vektor
n	[-]	nomalen Vektorkomponent
\mathbf{P}	[-]	Projektionsoperator
\mathbf{P}	[-]	Vorkonditionierermatrix
Pr	[-]	Produktion
Pr_l	[-]	Die laminare Prandtlzahl
Pr_t	[-]	Die turbulente Prandtlzahl
p	$[Pa]$	Drück des Strömungsmediums
\mathbf{Q}	[-]	Quelleterm
\mathbf{R}	[-]	Residuumsvektor
\mathbb{R}	[-]	positiv real nummer
Re	[-]	Reynolds-Zahl
\mathfrak{R}	$[J/(K \cdot mol)]$	die universelle Gaskonstante
\mathbf{S}	[-]	Glätter
S	[-]	Die Magnitude der Wirbelgröße
\tilde{S}	[m]	modifizierte Wirbelgröße
s	[-]	Der Zellstreckungskoeffizient
s	[-]	Stufe der implizites Runge-Kutta Verfahren

\bar{T}	[K]	Die Sutherland-Temperatur
t	[s]	Zeit
U	[-]	approximierte Lösung
U	[-]	obere Dreiecksmatrix
u	[m/s]	Geschwindigkeitskomponent des Strömungsmediums
\mathbf{V}	[-]	Korrektur
V	[m/s]	kontravariante Geschwindigkeit
\mathbf{W}	[-]	konservativen Variable
x, y, z	[m]	kartesische Koordinaten
z_i	[m ²]	horizontalen Fläche
α	[-]	Elemente der Koeffizientenmatrix der implizites Runge-Kutta Verfahren
α	[-]	Anstiegfaktor der CFL-Zahl
γ	[-]	Isentropenexponent
γ	[-]	Regularisierungsparameter
ΔT	[-]	Zeitschrittmatrix
Δt	[-]	Die Zeitschritt
ϵ	[-]	Das Relaxationparameter
ϵ_ψ	[-]	Koeffizient der Limiter
κ	[-]	Konduktion
κ_{eff}	[-]	effektive Konduktion
κ_l	[-]	laminare Konduktion
κ_t	[-]	turbulente Konduktion
λ	[Ns/m ²]	zweite Viskosität
λ_{eff}	[Ns/m ²]	effektive zweiten Viskosität
μ_{eff}	[Ns/m ²]	effektive Viskosität
μ_l	[Ns/m ²]	laminare Viskosität
μ_t	[m ² /s]	Wirbelviskosität
$\tilde{\nu}$	[m ² /s]	die kinematische Viskosität
ν_l	[m ² /s]	laminare kinematische Viskosität
ρ	[kg/m ³]	Dichte des Strömungsmediums
σ	[-]	ein Konstant für Spalart-Allmaras Turbulenzmodell
τ	[N/m ²]	Schubspannung
χ	[-]	spezifische kinematischen Viskosität
ψ	[-]	Limiter
Ω	[-]	offene Gebiet
$\omega_{M_{k,Pre}}$	[-]	Die Anzahl von pre-Glättung
$\omega_{M_{k+1}}$	[-]	Die Anzahl von Grobgitterglättung
$\omega_{M_{k,Post}}$	[-]	Die Anzahl von post-Glättung

1 Einleitung

Die Clean Sky Joint Undertaking (CSJU) hat Flightpath 2050 angenommen. Die Ziele der Flightpath 2050 sind: Im Jahr 2050 sind Flugzeuge in der Lage

- ihren Ausstoß um 75% von der heutigen CO₂ Emmission pro Kilometer und pro Passagierer zu reduzieren sowie 90% weniger NO_x-Emissionen zu produzieren.
- 65% der heutigen Lärmemissionen bei fliegenden Flugzeugen zu reduzieren
- die Zahl der allgemeinen Flugzeugunfälle im Vergleich zum Jahr 2000 um 80% zu senken.

Das heißt, dass die zukünftigen Flugzeuge ökologisch verbessert sind, weniger Lärm verursachen und viel sicherer sein sollen. Daher werden exakte und robuste Flugsimulationen durchgeführt, um diese Ziele zu erreichen. Zur Entwicklung neuer Flugzeuge werden viele experimentelle Tests mit vielen verschiedenen Parametern durchgeführt. In den letzten Jahren finden numerische Experimente aufgrund der raschen Entwicklung von Rechnern als Ergänzung oder Ersatz der Experimente in der Flugzeugindustrie und auch im akademischen Bereich immer mehr Anwendung. Die mathematischen Modelle werden mit numerischen Methoden dabei approximativ gelöst. Existierende Verfahren zum Lösen der mathematischen Modellgleichungen haben häufig viele Parameter, die von Benutzer gewählt werden müssen, und es ist häufig eine große Herausforderung, eine konvergierte Lösung zu bekommen.

Die Navier-Stokes Gleichung modelliert alle Phänomene aller Strömung. Jedoch ist die Gleichung wegen Komplexität nicht in der Lage, direkt für numerische Simulation zu verwenden. Vor allem ist es extrem schwer, eine numerische Lösung für turbulente Strömungen zu erhalten, da Turbulenz sowohl räumlich als auch zeitlich auf mehreren Skalen stattfindet. Um alle Phänomene zu simulieren, werden extrem feine Gitter und Zeitschritte benötigt. Die Simulation ist sogenannte die Direkte Numerische Simulation (DNS). Zur Überwindung des Problems werden turbulente Strömung durch mathematische Modelle vereinfacht. In dieser Arbeit wird nur der stationär Fall $\mathbf{R}(\mathbf{W}) = 0$ berücksichtigt, wobei \mathbf{R} Residuum und \mathbf{W} konservative Variable ist. Daher werden alle turbulenten Bewegungen modelliert und darf die turbulente Strömung durch Reynoldsdurchschnitte der Navier-Stokes Gleichung statistische modelliert werden. Solche Gleichungen nennt man daher auch Reynolds-gemittelte Navier-Stokes-Gleichungen (RANS). Anhand der kompressiblen Euler- und Navier-Stokes-Gleichungen werden in dieser Arbeit weitere Diskussionen angestellt 2.1.1.

Zum Lösen der RANS-Gleichungen wird eine Mehrgitter-Verfahren verwendet. Mehrgitter-Verfahren sind bekannt zur Beschleunigung der Konvergenz durch Glättung der Fehlern auf Grobgitterstufen und Glättung der kurzwelligen Fehlern auf der Feingitterstufe, mehr detail in Kapitel 3.1 gefunden werden. Explizite Runge-Kutta Verfahren wurden als Glätter in Mehrgitter Verfahren ursprünglich zum Lösen der kompressiblen Euler-Gleichung seit mehr als 30 Jahren verwendet [1]. Diese Glättungsmethode wurde auch zur Lösen der RANS Gleichung angewendet. Explizite Runge-Kutta Verfahren haben Beschränkungen in der CFL -Zahl und daher bringt das langsame Konvergenz oder keine Konvergenz.

Yoon und Jameson haben ein implizites Schema für strukturierte Diskretisierungen implementiert, das robust für einen großen Mach-Zahl Bereich ist und auf LU Faktorisierung basiert [2]. Außerdem

verbraucht das Schema wenig Speicher, das vor allem für große Testfälle ein Vorteil ist. Versuche haben gezeigt, dass das Schema im Allgemein robuster und effizienter als explizite Runge-Kutta Verfahren sind.

Es läßt sich häufig beobachten, dass bei der Berechnung der kompressiblen RANS-Gleichung in viskosen Fluß mit hohen Reynolds-Zahlen sehr langsam konvergiert und die Robustheit drastisch verloren geht. Ein Grund hierfür ist, dass die Zellen der Berechnungsgitter in der Grenzschicht sehr hohe Streckungen haben und in Strömungsrichtung gestreckt sind. Die folgende Abbildung 1.1 zeigt ein Beispiel einer gestreckte Zelle in der Grenzschicht.

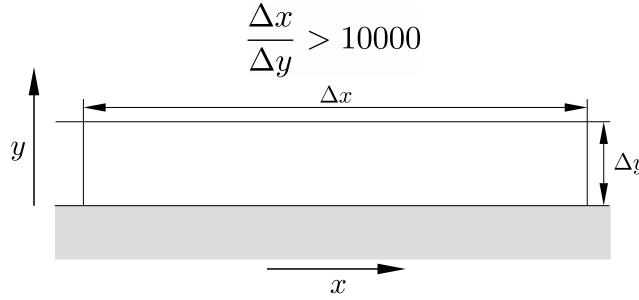


Abb. 1.1: Gedehte Zelle an der Grenzschicht

Diese gestreckten Zellen, die ein Streckungsverhältnis von $\frac{\Delta x}{\Delta y} > 10000$ haben können, verursachen eine Steifigkeit des diskretisierten Systems in den Gleichungen. Die Steifigkeit führt zu Schwierigkeiten bei der Glättung vom Fehlern im numerischen Verfahren. Pierce und Giles haben die numerische Steifigkeit im Artikel [3] diskutiert. Das bedeutet, dass explizite Verfahren als Glätter eine Mehrgitterverfahren nicht geeignet ist. Eine Idee zur Überwindung dieser Schwierigkeit ist es ein implizites Schema als Glätter zu verwenden. Für 2-D Strömungen haben Rossow, Swanson und Turkel erhebliche Fortschritt für Effizienz und Robustheit erlaubt, indem explizite mehrstufige Runge-Kutta Verfahren mit implizite Schema ergänzt wird [4][5][6]. In [7] wird das Baldwin-Lowmax Turbulenzmodell für 3-D Flügetest verwendet und zeigte Effizient der impliziten Glätter über Konvergenz. Die Punkt-implizite Runge-Kutta Verfahren wird von Langer implementiert und zeigte erhebliche Verbesserung der Konvergenz für nicht-viskosen stationäre Strömung [8]. Effizient und Robust der approximierten stationären Lösung durch Liner-implizit Verfahren für die Glätter der Agglomerierter Mehrgitter Verfahren hat Langer gezeigt [9]. In der Artikel ein-Gleichungstyp Spalart-Allmaras Turbulenz Modelle wird für 3-D Flügetest demonstriert. Der Hierarchie der Precondition Technik für mehr stufige Runge-kutta Verfahren um goben Gitter durch Grob-Gitter Gleichung für hohe Reynolds Zahl viskosen Strömung zu konstruieren ist in [10]. In [11] wird die Zusammenfassung der häufigen verwendeten impliziten Verfahren gefunden. Die meisten Investigationen der Entwicklung der Lösungsmethode sind Approximation der Jacobian und Lösungsmethode für lineare Gleichungssystem, welche in der Gätter approximative gelöst werden muss.

Das Turbulenzmodell in dieser Arbeit ist ein Ein-Gleichungs Spalart-Allmaras Modell für 3-D Testfall verwendet, da generell die mehr-Gleichungs turbulente Modelle zum Beispiel mit LU-SGS Verfahren direkt wegen Simplität des Konstruktion des Glätters implementiert werden kann. Nach Diskretisierung der RANS Gleichung erhalten wir ein System gewöhnlicher Differential-Gleichungen, welche der mittleren Strömungs Teil und der turbulente Strömungs Teil getrennt dargestellt werden kann. Das Systemen werden sich gekoppelt und zusammen gelöst werden. Das heißt, dass turbulente Variable konstant während Lösen der mittleren Strömungs Gleichung angenommen wird und umgekehrt.

In Kapitel 3.1 werden nicht-lineare Mehrgitterverfahren diskutiert. In der Arbeit werden mehrstufige

implizite Runge-Kutta Verfahren als Glätter für ein FAS (Full-Approximation Storage) Mehrgitter Verfahren, welches auf einer Approximation der Ableitung basiert, verwendet. Wenn man $\mathbf{CFL} = \infty$, ein Stufig und exakte Ableitung $\frac{d\mathbf{R}}{d\mathbf{W}}$ berücksichtigt, kann die impliziten Runge-Kutta Verfahren formuliert werden und die finale Formulierung ist nicht anderes als Newton-Verfahren. Hierfür kommt die Idee her, dass die Newton-Verfahren mit Levenberg-Marquardt Algorithmus verbessert werden kann. Ein Vorteil eines Levenberg-Marquardt Algorithmus zu verwenden lass die bessere Robustheit der Konvergenz sein werden, wird in Kapitel 3.3 vorstellt. Das Ziel dieser Arbeits ist es, die Ergebnisse zwischen diesen zwei Verfahren zu vergleichen und die Robustheit des Levenberg-Marquardt Algorithmus zu beurteilen.

In jedem Schritt des impliziten Runge-Kutta-Verfahrens muss ein lineares Gleichungssystem approximativ gelöst werden. Ein Symmetrisches (Block) Gauss-Seidel Verfahren wird dafür verwendet. Die Anzahl der inneren linearen Iterationen wird in dieser Arbeit 5 bis 25 getestet.

In dieser Arbeit werden NACA0012 Profil für 2-D Testfälle durch euler- und laminare Strömung demonstriert, die auf euler Gleichung und Navier-Stokes Gleichung ohne Aktivierung des viskosen Terms basieren. Für 3-D Testfälle wird Hemisphier-Zylinder mit turbulente Strömung getestet. Jeweils der Testfällen werden durch Newton-Verfahren und Levenberg-Marquardt Algorithmus mit singulär und parallel Domänen Berechnung durchführt.

2 Versuchsanordnung bzw. verwendete Software

Als Basiscode für die Entwicklungen in dieser Arbeit wird die Entwicklerversion des TAU-Codes verwendet. Der TAU-Code ist ein sowohl national als auch international weit verbreitetes und anerkanntes Werkzeug zur Analyse und Vorhersage aerodynamischer Fragestellungen, welches von mehr als 60 Wissenschaftlern am Institut für Aerodynamik und Strömungstechnik am DLR (Deutsches Zentrum für Luft- und Raumfahrttechnik) entwickelt wird. Dieses Werkzeug ermöglicht die Simulation viskoser Strömungen um komplexe Geometrien. Die unterstützte Bandbreite reicht von sehr langsamer bis hin zu hypersonischer Anströmung, sodass die Umströmung von Raumfahrzeugen beim Wiedereintritt in die Erdatmosphäre bis hin zu Flugzeugen im Start und Landevorgang simuliert werden kann. Im Kapitel Numerisches Beispiel 4 werden die Tests auf einem Hochleistungsrechner durchgeführt.

2.1 Grundlegende Gleichungen

2.1.1 Reynolds-gemittelte Navier-Stokes-Gleichungen

Turbulente Strömungen weisen sehr komplexe Strukturen auf und eine Simulation turbulenter Strömungen ist schwierig. Im Allgemeinen werden hierfür sehr viele Gitterzellen und sehr lange Rechenzeiten benötigt, sodass die direkte numerische Simulation (DNS) für hohe Reynoldszahlen fast unmöglich ist. Zur Erfassung turbulenter Strömungen wird daher eine statistische Modellierung angesetzt, welche die instationäre Strömung unter Verwendung eines zeitlichen Mittelwerts als stationäre Strömung behandelt. Es müssen jedoch die hierbei entstehenden nichtlinearen Zusammenhänge berücksichtigt werden. Die Navier-Stokes-Gleichungen beschreiben viskose Strömungen anhand einer nichtlinearen, partiellen Differentialgleichung zweiter Ordnung. In dieser Arbeit werden die dreidimensionalen, Reynolds-gemittelten Navier-Stokes-Gleichungen (RANS) zugrunde gelegt, welche auf einem offenen Gebiet $\Omega \subset \mathbb{R}^3$ definiert sind und unter Verwendung der konservativen Variable $\mathbf{W} := (\rho, \rho u_1, \rho u_2, \rho u_3, \rho E)$ wie im integralen Form folgt lauten.

$$\frac{d}{dt} \int_{\Omega} \mathbf{W} d\mathbf{x} + \int_{\partial\Omega} (\mathbf{f}_c \cdot \bar{\mathbf{n}} - \mathbf{f}_v \cdot \bar{\mathbf{n}}) d\mathbf{s} = 0 \quad (2.1)$$
$$\mathbf{f}_c \cdot \bar{\mathbf{n}} = \begin{pmatrix} \rho V \\ \rho u_1 V + p n_1 \\ \rho u_2 V + p n_2 \\ \rho u_3 V + p n_3 \\ \rho H V \end{pmatrix}, \mathbf{f}_v \cdot \bar{\mathbf{n}} = \begin{pmatrix} 0 \\ n_1 \tau_{11} + n_2 \tau_{21} + n_3 \tau_{31} \\ n_1 \tau_{21} + n_2 \tau_{22} + n_3 \tau_{32} \\ n_1 \tau_{31} + n_2 \tau_{23} + n_3 \tau_{33} \\ n_1 \theta_1 + n_2 \theta_2 + n_3 \theta_3 \end{pmatrix}$$

Hier ist $\mathbf{f}_c \cdot \bar{\mathbf{n}}$ der konvektive Fluss und $\mathbf{f}_v \cdot \bar{\mathbf{n}}$ der viskose Fluss. $\bar{\mathbf{n}} = (\bar{n}_1, \bar{n}_2, \bar{n}_3)$ bezeichnet den Normalenvektor bzw. seine Komponenten und V ist die kontravariante Geschwindigkeit, welche die Geschwindigkeit senkrecht zu dS bezeichnet und folgendermaßen definiert wird:

$$V = \langle \mathbf{u}, \bar{\mathbf{n}} \rangle = n_1 u_1 + n_2 u_2 + n_3 u_3. \quad (2.2)$$

Die totale Enthalpie H ist gegeben durch:

$$H(\mathbf{x}, t) = (\gamma - 1)\rho(\mathbf{x}, t) + \frac{p(\mathbf{x}, t)}{\rho(\mathbf{x}, t)}. \quad (2.3)$$

Der Druck p ist durch die Zustandsgleichung gegeben,

$$p(\mathbf{x}, t) = (\gamma - 1)\rho(\mathbf{x}, t) \left(E(\mathbf{x}, t) - \frac{\|\mathbf{u}\|_2^2}{2} \right) \quad (2.4)$$

wobei γ den Isentropenexponent bezeichnet, welcher für Luft den Wert 1,4 hat. Der viskose Term wird folgendermaßen definiert:

$$\tau_{ii} := 2\mu_{\text{eff}} \frac{\partial u_i}{\partial x_i} + \lambda_{\text{eff}} \text{div} u, \quad \lambda_{\text{eff}} := -\frac{2}{3}\mu_{\text{eff}}, \quad i = 1, 2, 3, \quad (2.5)$$

$$\tau_{ij} := \mu_{\text{eff}} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \quad \tau_{ij} := \tau_{ji}, \quad 1 \leq i < j \leq 3, \quad (2.6)$$

$$\theta_j = \left(\sum_{k=1}^3 u_k \tau_{jk} \right) + q_j, \quad j = 1, 2, 3, \quad (2.7)$$

$$\mathbf{q} := \kappa_{\text{eff}} \text{grad } T, \quad \mathbf{q} = (q_1, q_2, q_3). \quad (2.8)$$

Die effektive Viskosität und Konduktion in den Gleichungen (2.5) bis (2.7) ist gegeben durch:

$$\mu_{\text{eff}} := \mu_l + \mu_t, \quad \kappa_{\text{eff}} := \kappa_l + \kappa_t. \quad (2.9)$$

Wirbelviskosität μ_t und κ_t werden jeweils in (2.20) und (2.21) gegeben, und die laminare Viskosität μ_l wird durch Sutherlands Formel gegeben,

$$\mu_l := \mu_{l,\infty} \left(\frac{T}{T_\infty} \right)^{3/2} \frac{T_\infty + \bar{T}}{T + \bar{T}}, \quad \kappa_l := \frac{c_p \mu_l}{Pr_l} \quad \text{und} \quad c_p := \Re \epsilon \frac{\gamma}{\gamma - 1} \quad (2.10)$$

wobei $\bar{T} := 110.4K$ die Sutherland-Temperatur, $\Re \epsilon$ die universelle Gaskonstante und $Pr_l := 0.72$ die laminare Prandtlzahl ist.

2.1.2 Turbulenzmodelle

In diesem Abschnitt wird ein Turbulenzmodell vorgestellt, welche in 3-D Testfall dieser Arbeit verwendet wird.

Negatives Spalart-Allmaras Eingleichungs-Turbulenzmodell

Als negative Spalart-Allmaras Eingleichungs-Turbulenzmodell wird häufig verwendet. Das Turbulenzmodell wurde von Spalart und Allmaras im Jahre 1992 [12] vorgestellt. Die Transportgleichung für Spalart-Allmaras Turbulenzmodell ist gegeben durch,

$$\frac{d}{dt} \int_{\Omega} \tilde{\nu} d\mathbf{x} + \int_{\partial\Omega} (\mathbf{f}_{c,turb} \cdot \mathbf{n} - \mathbf{f}_{v,turb} \cdot \mathbf{n}) ds = \int_{\Omega} \mathbf{Q} d\mathbf{x} \quad (2.11)$$

wobei $\mathbf{f}_{c,turb}$ der turbulente konvektive und $\mathbf{f}_{v,turb}$ der turbulente viskose Fluss ist. Die Definition des konvektiven Flusses ist

$$\mathbf{f}_{c,turb} = \tilde{\nu} V, \quad (2.12)$$

Der turbulente viskose Fluss für $\tilde{\nu} \geq 0$ ist gegeben durch:

$$\mathbf{f}_{v,turb} = \left(\frac{\nu_l + \tilde{\nu}}{\sigma} \right) \left(\sum_{j=1}^3 n_j \frac{\partial \tilde{\nu}}{\partial x_j} \right) \quad (2.13)$$

Wenn $\tilde{\nu}$ kleiner als Null ist, laut der turbulente viskosen Fluss,

$$\mathbf{f}_{v,turb} = \left(\frac{\nu_l + f_n \tilde{\nu}}{\sigma} \right) \left(\sum_{j=1}^3 n_j \frac{\partial \tilde{\nu}}{\partial x_j} \right), \quad (2.14)$$

wobei gilt,

$$f_n = \frac{c_{n1} + \chi^3}{c_{n1} - \chi^3}, \quad c_{n1} := 16. \quad (2.15)$$

Der Quellterm von Gleichung (2.11) besteht aus einer einfachen Summe der Produktionsterm Pr , Destruktionsterm De und Diffusionsterm Di ,

$$Q := Pr + De + Di, \quad (2.16)$$

wobei, Produktionsterm Pr ist gegeben durch:

$$Pr := \begin{cases} c_{b1}(1 - f_{t2})\tilde{S}\tilde{\nu} & (\tilde{\nu} \geq 0) \\ c_{b1}(1 - c_{t3})\tilde{S}\tilde{\nu} & (\tilde{\nu} < 0) \end{cases}, \quad f_{t2} := c_{t3} \exp(-c_{t4}\chi^2), \quad (2.17)$$

Destruktionsterm De ist

$$De := \begin{cases} \left(c_{w1} f_w - \frac{c_{b1}}{\kappa^2} f_{t2} \left(\frac{\tilde{\nu}}{d} \right)^2 \right) & (\tilde{\nu} \geq 0) \\ -c_{w1} \left(\frac{\tilde{\nu}}{d} \right)^2 & (\tilde{\nu} < 0) \end{cases}, f_w := g \left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right)^{\frac{1}{6}}, \quad (2.18)$$

und Diffusionsterm Di ist

$$Di := \frac{c_{b2}}{\sigma} \sum_{j=1}^3 \left(\frac{\partial \tilde{\nu}}{\partial \chi_j} \right)^2. \quad (2.19)$$

Die Wirbel-Viskosität μ_t ist mit der Formel gegeben:

$$\mu_t := \begin{cases} \rho \tilde{\nu} f_{v1} & (\tilde{\nu} \geq 0) \\ 0 & (\tilde{\nu} < 0) \end{cases} \quad (2.20)$$

wobei

$$\kappa_t := \frac{c_p \mu_t}{P r_t}, \quad f_{v1} := \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi := \frac{\tilde{\nu}}{\nu_l}, \quad \nu_l := \frac{\nu_l}{\rho} \quad (2.21)$$

Hierbei wurden für die Konstanten in den Gleichungen (2.11) bis (2.20) nach [13] folgende Werte eingesetzt:

$$S := \|\text{curl}(\mathbf{u})\|_2, \quad \bar{S} := \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}, \quad \tilde{S} := \begin{cases} S + \bar{S}, & \bar{S} \geq -c_{v2} S \\ S + \frac{S(c_{v2}^2 S + c_{v3} \bar{S})}{(c_{v3} - 2c_{v2})S - \bar{S}}, & \bar{S} < -c_{v2} S, \end{cases} \quad (2.22)$$

$$g := r + c_{w2} r (r^5 - 1), \quad \min \left\{ \frac{\tilde{\nu}}{\kappa^2 d^2 \tilde{S}}, 10 \right\}, \quad f_{v2} := 1 - \frac{\chi}{1 + \chi f_{v1}} \quad (2.23)$$

$$c_{b1} = 0,1355, \quad c_{b2} = 0,622, \quad c_{v1} = 7,1, \quad c_{v2} = 0,7, \quad c_{v3} = 0,9 \quad (2.24)$$

$$\sigma = 2/3, \quad \kappa = 0,41, \quad c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{(1 + c_{b2})}{\sigma}, \quad (2.25)$$

$$c_{w2} = 0,3, \quad c_{w3} = 2, \quad c_{t3} = 1,2, \quad c_{t4} = 0,5 \quad (2.26)$$

d ist die Wandabstand. Die Freiströmung $\tilde{\nu}$ wird $3\nu_{l,\infty}$ eingesetzt, wobei die $\nu_{l,\infty}$ kinematische Viskosität in der Freiströmung ist [13].

2.2 Diskretisierung

Zur numerischen Rechnung werden die Gleichungen (2.1) und (2.11) mit einem finite-Volumen-Verfahren diskretisiert.

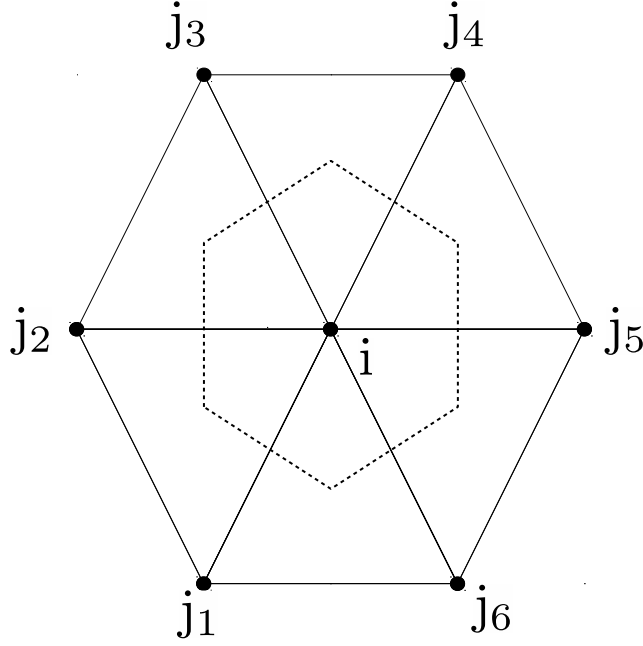


Abb. 2.1: Ein Beispiel des Dualgitters (gestrichelte Linie) und dreieckigen Primärgitters (durchgehende Linie)

Das Dualgitter wird in einem Preprocessing-Schritt erstellt. In Abbildung 2.1 zeigt ein Beispiel des Dualgitters. $G := 1, \dots, N$ beschreibt die Menge aller Punkte im Dualgitter mit unbekannten Zuständen $(\mathbf{W}_1, \tilde{\nu}_1), \dots, (\mathbf{W}_N, \tilde{\nu}_N)$. Die Nachbarnpunkte von $i \in G$ werden mit $N(i)$ bezeichnet und die zugehörige Oberfläche zwischen i und j mit ij . Werte auf der Oberfläche werden mit diesem Doppelindex gekennzeichnet. Beispielsweise sind die zu einem Oberflächenbereich zugehörigen Normalenvektoren, Normalgeschwindigkeiten und Schallgeschwindigkeiten gegeben durch:

$$\text{svol}(\Omega_{ij}) := \int_{\Omega_i \cap \Omega_j} 1 ds, \mathbf{n}_{ij} := \text{svol}(\Omega_{ij})(n_{1,ij}, n_{2,ij}, n_{3,ij}) \quad (2.27)$$

$$V_{ij} := \frac{1}{2} \langle (\mathbf{u}_i + \mathbf{u}_j), \mathbf{n}_{ij} \rangle, a_{ij} := \frac{1}{2} (a_i + a_j) \quad (2.28)$$

Der reibungsfreie Term aus Gleichung (2.1) wird mit einem Zentrale-Differenzen-Schema mit künstlicher Viskosität diskretisiert. Zum Abbilden der Stoßwelle wird ein druckabhängiger “Umschaltterm” in den dissipativen Term eingefügt. Dieser reduziert die Ordnung bei den Nachbarn der Stoßwelle auf eins. Zur Behandlung hoher gestreckter Gitter wird ein Zellstreckungskoeffizient eingeführt. Der reibungsfreie Term $\mathbf{f}_c \cdot \mathbf{n}$ für ein inneres Volumen Ω_i wird durch

$$\int_{\partial\Omega_i} \mathbf{f}_c \cdot \mathbf{n} ds \approx \sum_{j \in N(i)} \frac{1}{2} \left((\mathbf{f}_c \cdot \mathbf{n}_{ij})(\mathbf{W}_i) + (\mathbf{f}_c \cdot \mathbf{n}_{ij})(\mathbf{W}_j) \right) - \mathbf{D}_{ij}(\mathbf{W}), \quad (2.29)$$

$$\mathbf{D}_{ij}(\mathbf{W}) := \frac{1}{2} |\mathbf{A}_{ij}^{\text{Roe}}| \{ \psi_{ij}(\mathbf{W}_j - \mathbf{W}_i) - \xi_{s_{ij}}(\mathbf{W})(1 - \Psi_{ij})(L_j(\mathbf{W}) - L_i(\mathbf{W})) \} \quad (2.30)$$

approximiert, hierbei ist:

$$L_i(\mathbf{W}) := \sum_{j \in \mathcal{N}(i)} (\mathbf{W}_j - \mathbf{W}_i), \quad \mathbf{A}_{ij}^{\text{Roe}} := \frac{\partial(\mathbf{f}_c \cdot \mathbf{n}_{ij})}{\partial \mathbf{W}} [\mathbf{W}_{\text{Roe}}], \quad (2.31)$$

$$\Psi_{ij} := \min \{ \epsilon_\psi \Psi_{ij}^*, 1 \}, \quad \Psi_{ij}^* := \frac{(p_j - p_i)^2}{(p_j + p_i)^2}, \quad \epsilon_\psi := 8. \quad (2.32)$$

\mathbf{W}_{Roe} bedeutet, dass die Variable, die Roe-Durchschnitt durchgeführt wird, auf der Oberfläche ij verwendet wird, Matrix zu konstruieren. Bei der Konstruktion der Roe Matrix $\mathbf{A}_{ij}^{\text{Roe}}$ und der Entropy-fix vermeiden Null Dissipation [8]. Der Zellstreckungskoeffizient basiert auf dem Betrag des größten konvektiven Eigenwerts. Dieser wird durch

$$\lambda_i := \sum_{j \in \mathcal{N}(i)} \lambda_{ij}, \quad \lambda_{ij} := |\mathbf{V}_{ij}| + \text{svol}(\Omega_{ij}) a_{ij}, \quad (2.33)$$

$$s_{ij}(\mathbf{W}) := 1 + 2 \frac{\sqrt{z_i} \sqrt{z_j}}{\sqrt{z_i} + \sqrt{z_j}}, \quad z_i := \frac{\lambda_i - \lambda_{ij}}{\lambda_{ij}} \quad (2.34)$$

berechnet.

dh				
dh		k_4		
dh	k_3	i	k_1	
dh		k_2		
dh				
	dH	dH	dH	dH

Abb. 2.2: Beispiel eines anisotropen Gitters

Es wird eine Grenzschichtströmung in Wandnähe betrachtet, um den Ausdruck für s_{ij} herzuleiten. Es wird angenommen, dass der Betrag der Normalgeschwindigkeit in einer viskosen Grenzschicht viel kleiner als der Betrag der Schallgeschwindigkeit $a_{ik} \gg V_{ik}$ ist. Außerdem wird die Annahme getroffen, dass die Schallgeschwindigkeit konstant zwischen Nachbarzellen und $dH \gg dh$ ist. Zum Beispiel, zwischen i und k_1 für Senkrechts oberfläche wird das durch

$$z_i = \frac{\left\{ \sum_{k \in \mathcal{N}(i)} (|\mathbf{V}_{ik}| + \text{svol}(\Omega_{ik}) a_{ik}) \right\} - (|\mathbf{V}_{ik}| + \text{svol}(\Omega_{ik_1}) a_{ik_1})}{|\mathbf{V}_{ik_1}| + \text{svol}(\Omega_{ik_1}) a_{ik_1}} \quad (2.35)$$

$$\approx \frac{adH - dh}{dh} \approx 2 \frac{dH}{dh} \quad (2.36)$$

approximiert. Und das heißt,

$$s_{ij}(\mathbf{W}) \approx 1 + 2 \frac{2 \frac{dH}{dh}}{2 \sqrt{2 \frac{dH}{dh}}} = 1 + \sqrt{2} \sqrt{\frac{dH}{dh}}. \quad (2.37)$$

Durch Austausch der Rolle von dh und dH in Abbildung 2.2 bekommen wir für horizontalen Fläche (zwischen i und k_4)

$$z_i \approx \frac{2dH - dH}{dH} \approx 1, s_{ij}(\mathbf{W}) \approx 1 + 2 \frac{1}{2} = 2. \quad (2.38)$$

Der Wert s_{ij} in der Approximation aus Gleichung (2.37) ist dafür ausgelegt, die Dissipation für stark gestreckte Zellen in Richtung der großen Ausdehnung zu erhöhen. Dadurch verbessert sich die Zuverlässigkeit der Lösung. Der konvektive Fluss in der Gleichung (2.11) wird mit einem Roe-Schema erster Ordnung diskretisiert:

$$\int_{\partial\Omega_i} \mathbf{f}_{c,turb} \cdot \mathbf{n} ds \approx \sum_{j \in \mathcal{N}(i)} \frac{1}{2} \{ (\mathbf{f}_{c,turb} \cdot \mathbf{n}_{ij}) (\tilde{v}_i) + (\mathbf{f}_{c,turb} \cdot \mathbf{n}_{ij}) (\tilde{v}_j) \} - \frac{1}{2} |V| (\tilde{v}_j - \tilde{v}_i). \quad (2.39)$$

Hierbei wird für den dissipativen Term der Approximation aus Gleichung (2.39) $\frac{1}{2} |V| (\tilde{v}_j - \tilde{v}_i)$ kein Entropie Fix verwendet. Zur Diskretisierung des viskosen Terms $\mathbf{f}_v \cdot \mathbf{n}$ aus Gleichung (2.1), $\mathbf{f}_{v,turb} \cdot \mathbf{n}$ aus Gleichung (2.11) wird die Green-Gauß-Methode verwendet. Ausgedrückt in Bezug auf eine allgemeine Variable w erhält man damit:

$$\frac{\partial w_i}{\partial x_k} \approx \left(\frac{\partial w_i}{\partial x_k} \right)^{\text{GG}} := \frac{1}{\text{vol}(\Omega_i)} \sum_{j \in \mathcal{N}(i)} \text{svol}(\Omega_{ij}) \frac{n_{k,ij}}{2} (w_i + w_j), k = 1, 2, 3, \quad (2.40)$$

Alternativ erhält man unter Vernachlässigung um Richtungen:

$$\frac{\partial w_{ij}}{\partial x_k} \approx \left(\frac{\partial w_{ij}}{\partial x_k} \right)^{\text{TSL}} := n_{k,ij} \frac{(w_j - w_i)}{\|\mathbf{x}_i - \mathbf{x}_j\|_2} \quad (2.41)$$

Die Green-Gauß-Methode (2.40) liefert einen Gradienten am Punkt im Kontrollvolumen i , die unter Vernachlässigung um Richtungen (2.41) dagegen einen Gradienten auf der Fläche. Mit der Green-Gauß-Methode wird auf der gesuchten Fläche Ω_{ij} ein Gradient von

$$\frac{\partial w_{ij}}{\partial x_k} := \frac{1}{2} \left[\left(\frac{\partial w_i}{\partial x_k} \right)^{\text{GG}} + \left(\frac{\partial w_j}{\partial x_k} \right)^{\text{GG}} \right] \quad (2.42)$$

berechnet. Durch Mittelung der Geschwindigkeit, Viskosität, Konduktivität in viskosen Term werden $\mathbf{f}_v \cdot \mathbf{n}$ und $\mathbf{f}_{v,turb} \cdot \mathbf{n}$ mit Hilfe von der Gleichung

$$\int_{\partial\Omega_i} \mathbf{f}_v \cdot \mathbf{n} ds \approx \sum_{j \in \mathcal{N}(i)} (\mathbf{f}_v \cdot \mathbf{n}_{ij}) (\mathbf{W}_i, \mathbf{W}_j, \text{grad} \mathbf{W}_i, \text{grad} \mathbf{W}_j) \quad (2.43)$$

diskretisiert. Die Diskretisierung des Quellterms Q aus Gleichung (2.11) ergibt unter Verwendung des Green-Gauß-Gradienten:

$$\int_{\Omega_i} Q d\mathbf{x} \approx \text{vol}(\Omega_i) Q(\mathbf{W}_i, \text{grad} \mathbf{W}_i) \quad (2.44)$$

Diskretisierung der Randbedingungen

Alle erforderlichen Randbedingungen werden als Flüsse implementiert. Insbesondere wird gemäß [14] die Methode der sogenannten viskosen Wand verwendet. Gemäß Abbildung 2.3 wird der Geschwindigkeitsgradient an der Wand mit Diskretisierung (2.43) berechnet, wobei die Geschwindigkeit an der Wand zu null gesetzt wird:

$$\frac{\partial u_{\mathbf{P}_b}^{(l)}}{\partial x_k} \approx \left(\frac{\partial u_{\mathbf{P}_b}^{(l)}}{\partial x_k} \right)_{\text{bdry}}^{\text{TSL}} := n_{k, \mathbf{P}_n \mathbf{P}_b} \frac{-u_{\mathbf{P}_b}^{(l)}}{\|\mathbf{X}_{\mathbf{P}_b} - \mathbf{X}_{\mathbf{P}_n}\|_2}, \quad l, k = 1, 2, 3. \quad (2.45)$$

Der Geschwindigkeitsgradient an der Wand kann auch mit der Green-Gauß-Methode nach Gleichung (2.42) berechnet werden. Der Temperaturgradient an der Wand wird mit null angenommen, dies bedeutet, es wird von einer isothermen Wand ausgegangen. Der Druck wird mit einer Geschwindigkeit von null berechnet.

$$\frac{\partial T_{\mathbf{P}_b}^{(l)}}{\partial x_k}, \quad k = 1, 2, 3, \quad p_{\mathbf{P}_b} = (\gamma - 1)\rho E \quad (2.46)$$

Für die Turbulenzmodell-Gleichung (2.11) wird die viskose Wand wie oben auch als Flussbedingung implementiert. In diesem Fall wird $\tilde{\nu}_{\mathbf{P}_b} = 0$ angenommen. Auch der Quellterm wird zu null gesetzt. Fernfeldrandbedingung ist von Charakteristischen-Methode gegeben, wie es Freistromrandbedingung ist. Die erforderliche Eigenwertdekomposition des konvektiven Flusses kann in den Artikeln [8] und [9] nachvollgezogen werden.

Aus der Diskretisierung der grundlegenden Gleichungen (2.1) und (2.11) erhält man ein gewöhnliches Differentialgleichungssystem, welches folgendermaßen zusammengefasst werden kann:

$$\frac{d}{dt} \begin{pmatrix} \mathbf{W}(t) \\ \tilde{\nu}(t) \end{pmatrix} = \begin{pmatrix} -\mathbf{M}_{mean}^{-1} \mathbf{R}_{mean}(\mathbf{W}(t), \tilde{\nu}(t)) \\ -\mathbf{M}_{turb}^{-1} \mathbf{R}_{turb}(\mathbf{W}(t), \tilde{\nu}(t)) \end{pmatrix} \quad (2.47)$$

wobei $\mathbf{M}_{mean} := \text{diag}(\text{diag}(\text{vol}(\Omega_i))) \in \mathbb{R}^{5N \times 5N}$ und $\mathbf{M}_{turb} := \text{diag}(\text{vol}(\Omega_i)) \in \mathbb{R}^{N \times N}$ die Massenmatrizen für die Hauptströmung bzw. turbulente Gleichung sind. Da der stationäre Strömungsfall

betrachtet wird, verschwindet die zeitliche Ableitung und die vorige Gleichung kann umgeschrieben werden zu:

$$\begin{pmatrix} -\mathbf{M}_{mean}^{-1} \mathbf{R}_{mean}(\mathbf{W}(t), \tilde{\nu}(t)) \\ -\mathbf{M}_{turb}^{-1} \mathbf{R}_{turb}(\mathbf{W}(t), \tilde{\nu}(t)) \end{pmatrix} = 0. \quad (2.48)$$

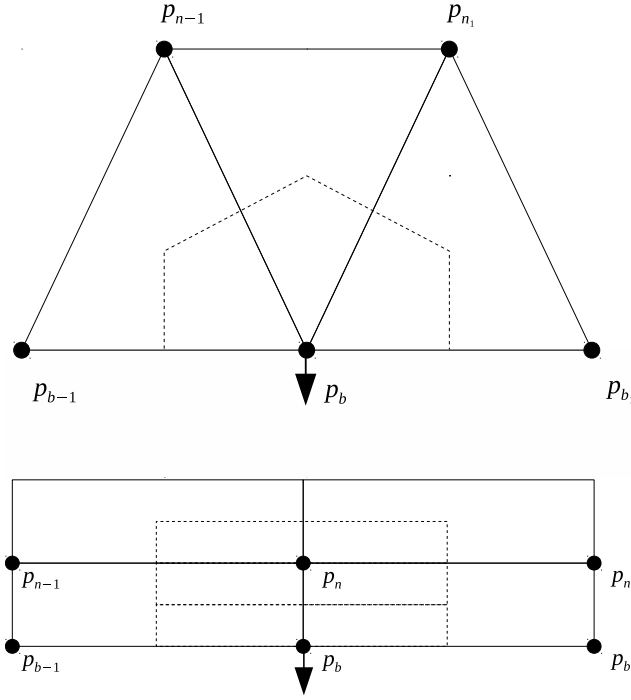


Abb. 2.3: Grenzpunkte des Dualgitters

wobei, \mathbf{R} entspricht der Residuumsvektor.

3 Numerische Lösungsmethode

Zum approximativen Lösen der nicht-linearen algebraischen diskretisierten System (2.48) ist ein effektives Lösungsverfahren notwendig. Bild 3.1 gibt einen Überblick über unsere Lösungsmethode. Dieses stellt die in jedem Schritt numerisch zu lösenden Gleichungen und die jeweils erforderlichen Ingredienzen dar. Im ersten Schritt wird das FAS-Mehrgitterverfahren (Full Approximation Scale) angewandt, welches in diesem Kontext ein agglomeriertes Mehrgitterverfahren ist. Es folgt das implizite Runge-Kutta-Verfahren als Glätter für das FAS-Mehrgitterverfahren. In der dritten Stufe wird das entstandene lineare System durch direkte Anwendung eines Newtonschen Verfahrens oder mit dem Levenberg-Marquardt-Algorithmus gelöst.

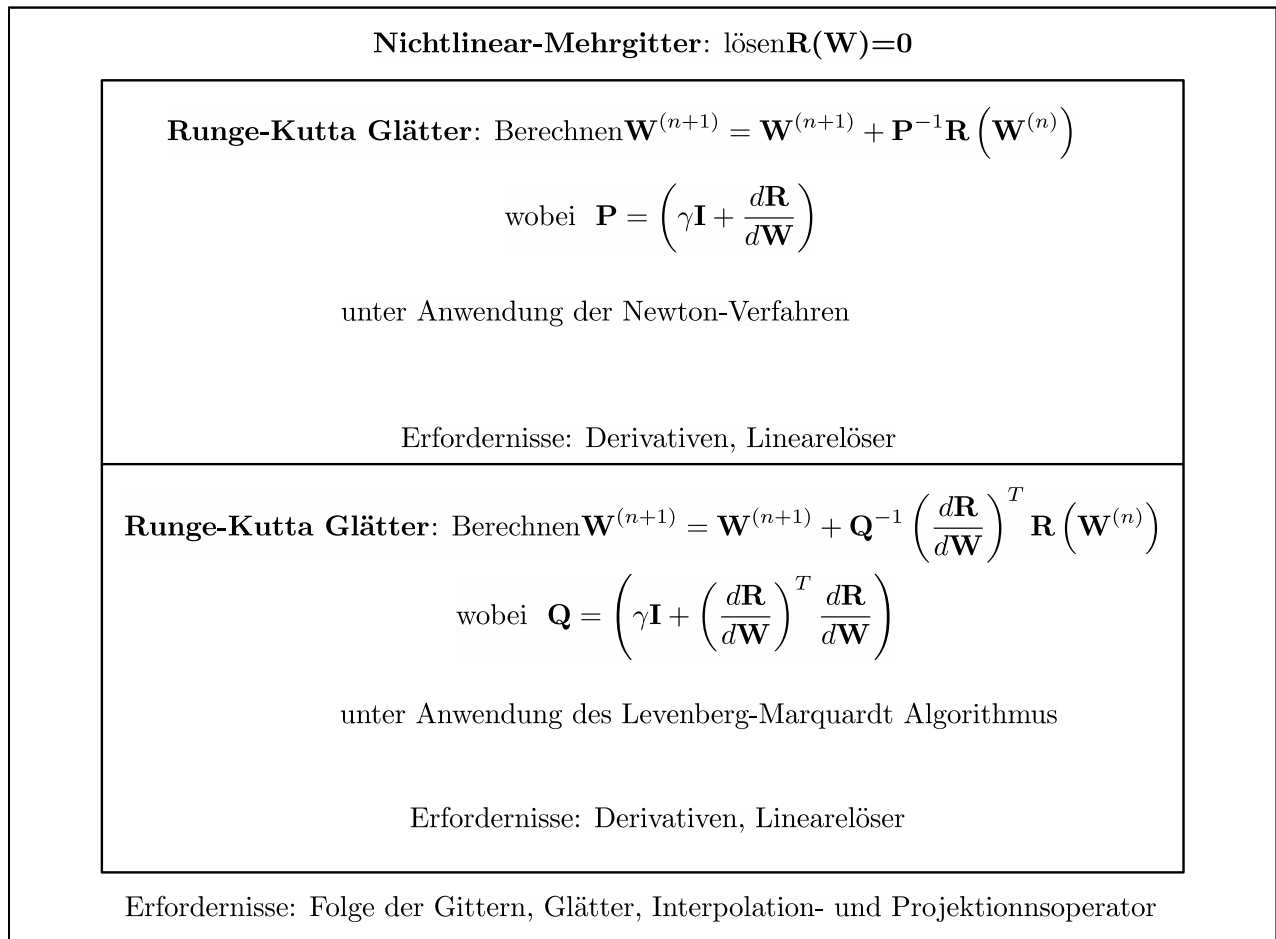


Abb. 3.1: Algorithmische Struktur der nichtlinearen Lösungsmethode

3.1 Nichtlineares Mehrgitterverfahren

Diese Methode ist bekanntermaßen effektiv zum Lösen großer nicht-linearer Gleichungssysteme. Im Allgemeinen hat die Iterationsmethode mit dem Mehrgitter das charakteristische Merkmal, dass das Residuum in der ersten Phase der Iteration schnell sinkt, danach jedoch wesentlich langsamer, da die kurzwelligen Fehlermoden in der ersten Phase sehr schnell geglättet werden, die langwelligen jedoch erhalten bleiben. Je größer die Zellenzahl des Gitters, desto stärker ist dieser Effekt ausgeprägt. Die Idee des Mehrgitterverfahrens ist es, eine Näherungslösung auf einem feinen Gitter auf ein gröberes Gitter zu projizieren, dort Fehlerkorrekturen zu berechnen und schließlich wieder auf das vorherige feine Gitter zu interpolieren, um die langwelligen Fehlermoden zu vermindern. Das nicht-lineare Mehrgitterverfahren wird in dieser Arbeit durch Aggregation realisiert. Im ersten Schritt muss das Grobgitter definiert werden. Anschließend wird der Projektions- und Interpolationsoperator formuliert, um die Daten von einer Gitterstufe auf die nächste zu transportieren. Zuletzt wird ein effektiver Glätter benötigt. Detaillierte Informationen über die Konstruktion der Grobgitterstufe kann in den Artikeln [15] und [11] gefunden werden. Der Mehrgitter-Algorithmus entstammt [16]. Das n -stufige Gitter M_n ist eine Folge von Gittern, die folgender Bedingung gehorcht:

$$M_n \subset \dots \subset M_1. \quad (3.1)$$

M_n wird für das begrenzte Gebiet $\Omega \subset \mathbb{R}^3$ und zur Formulierung des nichtlinearen vollen Approximationsschemas an die nichtlineare Gleichung $\mathbf{R}(\mathbf{W}) = \mathbf{f}$ berücksichtigt, wobei \mathbf{f} rechteitiger Vektor der nichtlinearen Gleichung und \mathbf{W} die Lösung sind. Hier sieht die nichtlineare Gleichung in ersten Gitterstufe M_1 :

$$\mathbf{R}_{M_1} : \mathbb{R}^{|M_1|} \rightarrow \mathbb{R}^{|M_1|}, \quad \mathbf{R}_{M_1}(\mathbf{W}_{M_1}) = \mathbf{f}_{M_1}. \quad (3.2)$$

Der Operator \mathbf{R}_{M_1} stellt beispielsweise die Diskretisierung der partiellen Differentialgleichung oder Integralgleichung auf ein gegebenes Gitter dar. Die Definitionen für einen Projektionsoperator P von einem feinen zum nächstgrößeren Gitter und für einen Interpolationsoperator I von einem groben zum nächstfeineren Gitter für $1 < k < n$ lauten:

$$\begin{aligned} P_{M_k}^{M_{k-1}} : \mathbb{R}^{|M_{k-1}|} &\rightarrow \mathbb{R}^{|M_k|}, \\ I_{M_{k-1}}^{M_k} : \mathbb{R}^{|M_k|} &\rightarrow \mathbb{R}^{|M_{k-1}|}. \end{aligned}$$

In der Integralgleichung (2.1) stellt \mathbf{R}_{M_1} außerdem ein System mit $d > 1$ Freiheitsgraden pro Kontrollvolumen dar. Dadurch wird die nichtlineare Gleichung (3.2) durch

$$\mathbf{R}_{M_1} : \mathbb{R}^{d|M_1|} \rightarrow \mathbb{R}^{d|M_1|}, \quad \mathbf{R}_1(\mathbf{W}_{M_1}) = \mathbf{f}_{M_1} \quad (3.3)$$

ersetzt. Die Lösung $\mathbf{W}_{M_1}^*$ aus (3.2) und (3.3) in der Iteration l kann mithilfe der Korrektur $V_{M_1}^{(l)}$ und einer gegebenen approximierten Lösung $U_{M_1}^{(l)}$ konstruiert werden:

$$\mathbf{W}_{M_1}^* = \mathbf{U}_{M_1}^{(l)} + \mathbf{V}_{M_1}^{(l)} \quad (3.4)$$

Dann erhält man die Defektgleichung auf dem Gitter M_1 aus

$$\mathbf{R}_{M_1} \left(\mathbf{U}_{M_1}^{(l)} + \mathbf{V}_{M_1}^{(l)} \right) - \mathbf{R}_{M_1}^{(l)} \left(\mathbf{U}_{M_1}^{(l)} \right) = \mathbf{d}_{M_1}^{(l)} \quad (3.5)$$

Dies erfordert, dass für $\mathbf{U}_{M_1}^{(l)} + \mathbf{V}_{M_1}^{(l)}$ gelöst wird. Es sei angemerkt, dass sich die Korrektur ergibt und den Defekt $\mathbf{V}_{M_1}^{(l)} = 0$ und $\mathbf{d}_{M_1}^{(l)} = 0$ erfüllen, wenn $\mathbf{U}_{M_1}^{(l)} = \mathbf{W}_{M_1}^*$ ist. Eine Approximation der Gleichung für das nächste Grobgitter M_2 ist:

$$\mathbf{R}_{M_2} \left(\mathbf{U}_{M_2}^{(l)} + \mathbf{V}_{M_2}^{(l)} \right) - \mathbf{R}_{M_2}^{(l)} \left(\mathbf{U}_{M_2}^{(l)} \right) = \mathbf{d}_{M_2}^{(l)} \quad (3.6)$$

wobei \mathbf{R}_{M_2} einer Approximation des nichtlinearen Operators \mathbf{R}_{M_1} entspricht. Die anderen unbekannten Operatoren in Gleichung (3.6) lauten:

$$\mathbf{U}_{M_2}^{(l)} := P_{M_2}^{M_1} \mathbf{U}_{M_1}^{(l)}, \quad (3.7)$$

$$\mathbf{V}_{M_2}^{(l)} := P_{M_2}^{M_1} \mathbf{V}_{M_1}^{(l)}, \quad (3.8)$$

$$\mathbf{d}_{M_2}^{(l)} := P_{M_2}^{M_1} \mathbf{d}_{M_1}^{(l)}. \quad (3.9)$$

Am Ende kann die Gleichung (3.6) nach $\mathbf{U}_{M_2}^{(l)} + \mathbf{V}_{M_2}^{(l)}$ gelöst werden:

$$\mathbf{R}_{M_2} \left(\mathbf{U}_{M_2}^{(l)} + \mathbf{V}_{M_2}^{(l),*} \right) = \mathbf{d}_{M_2}^{(l)} - \mathbf{R}_{M_2} \left(\mathbf{U}_{M_2}^{(l)} \right) \quad (3.10)$$

Die Korrektur für die defekte Gleichung (3.5) wird durch

$$\mathbf{U}_{M_2}^{(l+1)} := \mathbf{U}_{M_1}^{(l)} + \mathbf{V}_{M_2}^{(l)} \mathbf{U}_{M_2}^{(l)} \quad (3.11)$$

berechnet. Für gewöhnlich wird die defekte Grobgittergleichung (3.3) nicht exakt, sondern näherungsweise gelöst. Die neue approximierte Lösung $\mathbf{U}_{M_1}^{(l+1)}$ wird für den nächsten Glättungsschritt M_2 verwendet. Das nichtlineare Mehrgitterverfahren ist unten Algorithmus 1 zusammengefasst.

Dieser 2V-Zyklus von Algorithmus 1 ist der Prototyp des Mehrgitterverfahrens. Mit verschiedenen rekursiven Kombinationen werden in Abbildung 3.1 verschiedene Mehrgitter-Zyklen, wie beispielsweise der W- und F-Zyklus, formuliert.

Algorithm 1 Nichtlineares Mehrgitterverfahren

Angenommen, die Operatorgleichung für eine Folge der Gitter $M_n \subset \dots \subset M_1$ ist

$$\mathbf{R}_{M_k}(\mathbf{U}_{M_k}^{(l)}) = \mathbf{f}_{M_k}, k = 1, \dots, n, \quad (3.12)$$

und eine approximierte Lösung $\mathbf{U}_{M_k}^{(l)}$ mit Glätter $\mathbf{S} = \mathbf{S}(\mathbf{U}_{M_k}^{(l)}, \mathbf{R}_{M_k} \mathbf{f}_{M_k})$ sind gegeben. Hierbei ist \mathbf{S} eine Methode, welche einen neuen Iterationsschritt der approximierten Lösung $\mathbf{U}_{M_k}^{(l)}$,

$$\overline{\mathbf{U}}_{M_1}^{(l)} := \mathbf{S}(\mathbf{U}_{M_k}^{(l)}, \mathbf{R}_{M_k} \mathbf{f}_{M_k}) \quad (3.13)$$

berechnet. Es seien folgende Bezeichnungen definiert:

- $\omega_{M_k, Pre} \in \mathbb{N}_0$ Anzahl von pre-Glättungen
- $\omega_{M_{k+1}} \in \mathbb{N}_0$ die Anzahl von Grobgitterglättungen
- $\omega_{M_k, Post} \in \mathbb{N}_0$ Anzahl von post-Glättungen

Danach wird ein $2V = 2V(k, \omega_{M_k, Pre}, \omega_{M_{k+1}}, \omega_{M_k, Post})$ Mehrgitter-Zyklus durchgeführt:

- Glättungsschritte $\omega_{M_k, Pre}$ ausführen und folgende Gleichung auswerten

$$\overline{\mathbf{U}}_{M_k}^{(l)} = \mathbf{S}^{\omega_{M_k, Pre}}(\mathbf{U}_{M_k}^{(l)}, \mathbf{R}_{M_k} \mathbf{f}_{M_k}) \quad (3.14)$$

- die rechte Seite der Grobgittergleichung berechnen

$$\mathbf{f}_{M_{k+1}} = \mathbf{P}_{M_{k+1}}^{M_k} \left(\mathbf{f}_{M_k} - \mathbf{R}_{M_k} \left(\overline{\mathbf{U}}_{M_k}^{(l)} \right) \right) + \mathbf{R}_{M_k} \left(\mathbf{P}_{M_{k+1}}^{M_k} \overline{\mathbf{U}}_{M_k}^{(l)} \right) \quad (3.15)$$

- Grobgitterglättungsschritte $\omega_{M_{k+1}}$ ausführen und folgende Gleichung auswerten

$$\overline{\mathbf{U}}_{M_{k+1}}^{(l)} = \mathbf{S}^{\omega_{M_{k+1}}}(\mathbf{P}_{M_{k+1}}^{M_k} \mathbf{U}_{M_k}^{(l)}, \mathbf{R}_{M_{k+1}} \mathbf{f}_{M_{k+1}}) \quad (3.16)$$

- einen neuen Iterationsschritt der approximierten Lösung berechnen

$$\overline{\mathbf{U}}_{M_k}^{(l+1)} = \overline{\mathbf{U}}_{M_k}^{(l)} + \mathbf{I}_{M_{k+1}}^{M_k} \left(\overline{\mathbf{U}}_{M_{k+1}}^{(l+1)} - \mathbf{P}_{M_{k+1}}^{M_k} \overline{\mathbf{U}}_{M_{k+1}}^{(l+1)} \right) \quad (3.17)$$

- Glättungsschritte $\omega_{M_k, post}$ ausführen und folgende Gleichung auswerten

$$\mathbf{U}_{M_k}^{(l)} = \mathbf{S}^{\omega_{M_k, post}}(\overline{\mathbf{U}}_{M_k}^{(l+1)}, \mathbf{R}_{M_k} \mathbf{f}_{M_k}) \quad (3.18)$$

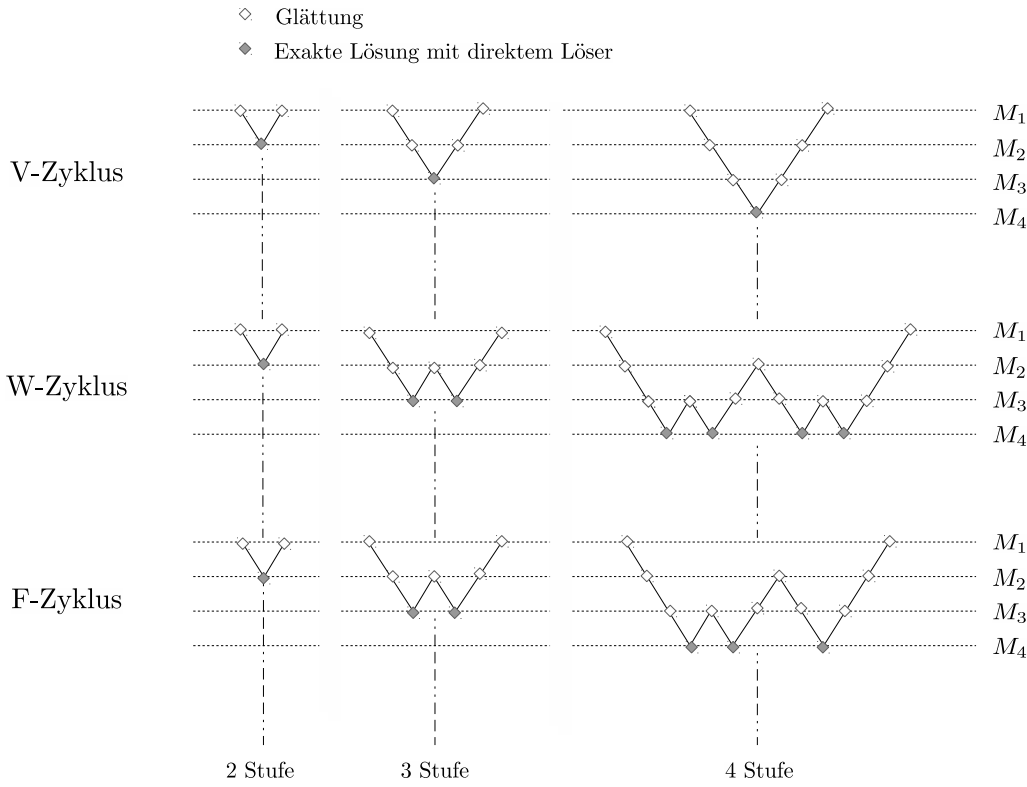


Abb. 3.2: Beispiel des Mehrgitterzyklus

3.1.1 Grobgittergleichung

Zur Implementierung der Grobgittergleichung müssen das Residuum, der numerische Fluss und die Weiterführung zum Feingitter berücksichtigt werden. Es gibt zwei Möglichkeiten zur Implementierung der Grobgitter-Gleichung,

1. Diskretisierung auf dem Grobgitter
2. Die Grobgitter-Gleichung.

Nun definieren wir die Punkte

$$\text{child}(C_i^{(k)}) := \{j \in \mathbf{G} : C_j^{(1)} \cap C_i^{(k)} \neq \emptyset\} \quad (3.19)$$

wobei $C_j^{(k)} \subset \Omega$ die i -te Grobgitterzelle auf dem Gitterlevel k ist. $k = 1$ entspricht dem Feingitter, $C_i^{(k)}$ wird durch $C_j^{(1)}$ zusammengefasst, wobei j zu allen Kindern von $C_i^{(k)}$ gehört:

$$C_j^{(1)} \subset C_j^{(1)}, \quad \bigcup_{j \in \text{child}(C_j^{(1)})} C_j^{(1)} = C_j^{(k)} \quad (3.20)$$

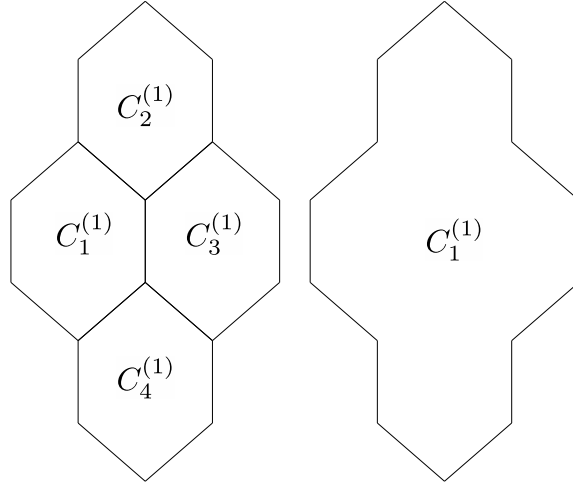


Abb. 3.3: Ein Skizze der Dualgitter mit vier Zellen (links) und ein Agglomeriate Zell (rechts)

Beispielweise wird ein grob Gitterzell auf dreieckige Gitter, die auf Agglomeration von vier Zellen basiert, in Abbildung 3.3 gezeigt. Zur Auswertung aller Flussterme wird nun das Volumen und der Normalenvektor der agglomerierten Zelle betrachtet. Die Beide werden über die metrischen Daten von Feingitter bestimmt. Danach werden die Grobgitter-Gleichungen diskretisiert, um die Korrektur des Grobgitters zu berechnen. Dabei wird die gleiche Strategie wie für das Feingitter verwendet. Die Flussvariable $\mathbf{W}_i^{(k)}$ gehört zur i -ten Zelle auf der k -ten Stufe. Diese wird durch einfache Injektionen zur feinsten Gitterstufe prolongiert. Die Zelle auf der feinsten Gitterstufe ist ein Kind der Zelle auf der größten Gitterstufe. \mathbf{W} ist von der groben Zelle gegeben durch:

$$\mathbf{W}_j^{(1),\text{corr}} := \mathbf{W}_i^{(k),\text{corr}}, \quad j \in \text{child}(C_i^{(k)}) \quad (3.21)$$

Der numerische Fluss des Grobgitters kann dadurch berechnet werden, dass alle Flüsse an allen Rändern der Zellen vom Feingitter summiert werden. Das Residuum der i -ten Zelle kann dann aus der Aufsummierung der Flüsse ausgewertet werden, die aus den Rändern der Feingitter-Zellen zu den Grobgitter-Zellen folgen.

$$\mathbf{R}_i^{(k)} = \sum_{j \in \text{child}(C_i^{(1)})} \sum_{k \in \mathbf{N}(j)} (\mathbf{f}_{kj} \cdot \mathbf{n}_{kj}) = \sum_{e \in \text{edges}(i)} (\mathbf{f}_e \cdot \mathbf{n}_e) \quad (3.22)$$

In Gleichung (3.22) erkennt man, dass zum Berechnen des Residuums nur die Ränder der Grobgitterzellen berücksichtigt werden, da der innere Fluss in der i -ten Zelle des Grobgitters verschwindet. Die Gleichung (3.22) kann für die konvektive-, viskosen- und Quellen-Termen in turbulenten Gleichung angewendet werden. Gradienten in der viskosen- und Quellen-Termen der Gleichung (2.1) werden dadurch auch berechnet. Wegen Inkonsistenz im viskosen Term, wird der Term mit der Faktor $(\frac{1}{2})^{k-1}$ gewichtet[17]. Auch die Berechnen der Preconditioner und der ΔT für grob Gitter wird durch gleichen Technik durchführt.

Letztendlich ist der Restriction Operator mit einem volumen gewichteten Interpolation gegeben durch:

$$\mathbf{W}_i^{(k)} = \frac{1}{\text{vol}(C_i^k)} \sum_{j \in \text{child}(C_i^{(k)})} \text{vol}(C_j^{(1)}) \mathbf{W}_j^{(1)} \quad (3.23)$$

3.2 Implizites Runge Kutta Verfahren

Implizite Verfahren werden als Glätter in einem Mehrgitterverfahren zum Lösen der Gleichung (2.1) verwendet. Generell können die Glätter als mehrstufige Runge-Kutta Verfahren formuliert werden. In dieser Arbeit wird vorzugsweise ein diagonal implizites Runge-Kutta Verfahren verwendet, um die diskretisierte Strömungsgleichungen (2.48) zu lösen. Diagonal implizit Runge-Kutta Verfahren kann mit dem Butcher Schema 3.1 vorstellt werden.

Tab. 3.1: Butcher Schema

$$\begin{array}{c|c} c & \mathcal{A} \\ \hline & b^T \end{array}$$

mit

$$\mathcal{A} := \begin{pmatrix} \alpha_{11} & 0 & \dots & 0 \\ \alpha_{21} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & \alpha_{s,s-1} & \alpha_{ss} \end{pmatrix}, \quad b := \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \alpha_{s+1,s} \end{pmatrix} \quad \text{und} \quad c := \begin{pmatrix} 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} \quad (3.24)$$

In Gleichung (3.24) sind die Koeffizienten des s -Stufigen Runge-Kutta-Verfahrens gegeben. Die Stufe des Schemas und die diskretisierte Evolution sind mit

$$\begin{aligned} k_1(\mathbf{W}) &:= -\mathbf{M}^{-1} \mathbf{R}(\mathbf{W}^{\mathbf{T}_n} + \alpha_{11} \Delta t k_1(\mathbf{W})) \\ k_2(\mathbf{W}) &:= -\mathbf{M}^{-1} \mathbf{R}(\mathbf{W}^{\mathbf{T}_n} + \alpha_{21} \Delta t k_1(\mathbf{W}) + \alpha_{22} \Delta t k_2(\mathbf{W})) \\ &\vdots \\ k_s(\mathbf{W}) &:= -\mathbf{M}^{-1} \mathbf{R}(\mathbf{W}^{\mathbf{T}_n} + \alpha_{s,s-1} \Delta t k_{s-1}(\mathbf{W}) + \alpha_{s,s} \Delta t k_s(\mathbf{W})) \end{aligned} \quad (3.25)$$

gegeben. Die Stufekoeffizient k_s braucht in jeden Stufe gelöst zu werden. Alle k_1, \dots, k_s werden in der nächsten Stufe verwendet, um

$$\mathbf{W}^{\mathbf{T}_{n+1}} = \mathbf{W}^{\mathbf{T}_n} + \alpha_{s+1} \Delta t k_s(\mathbf{W}) \quad (3.26)$$

zu bestimmen. Hier werden die Sub-Indizes *mean* und *turb* in den diskretisierten Strömungsgleichungen (2.48) nicht berücksichtigt, da die Lösungsmethode zur Lösen (2.48) für mittle und turbulente Strömung gleich ist. Generell sind die Gleichungen (3.25) nichtlinear. Daher ist es schwierig, Stufekoeffizienten k_1, \dots, k_s sofort gelöst zu werden, so dass ein Newton-Verfahren verwendet wird. Eine Lösung der nichtlinearen Gleichung sieht wie folgt aus:

$$g_j((k_j(\mathbf{W}))) := k_j(\mathbf{W}) + \mathbf{M}^{-1}\mathbf{R}(\mathbf{W}^{\mathbf{T}_n} + \alpha_{j,j-1}\Delta tk_{j-1}(\mathbf{W}) + \alpha_{jj}\Delta tk_j(\mathbf{W})), \quad (3.27)$$

nach nur einer Iteration abgebrochen wird. Die Ableitung der Gleichung (3.27) ist durch

$$\frac{dg_j(k_j(\mathbf{W}))}{dk_j(\mathbf{W})} = \mathbf{I} + \alpha_{jj}\Delta t\mathbf{M}^{-1}\frac{d\mathbf{R}}{d\mathbf{W}}(\mathbf{W}^{\mathbf{T}_n} + \alpha_{j,j-1}\Delta tk_{j-1}(\mathbf{W}) + \alpha_{jj}\Delta tk_j(\mathbf{W})). \quad (3.28)$$

gegeben. Ein Startwert ist durch $k_j^{(0)}(\mathbf{W}) = 0$ und eine Approximation dieser Lösung ist gegeben durch

$$k_j(\mathbf{W}) = -[\mathbf{P}_j(\mathbf{W})]^{-1}\left(g_j\left(k_j^{(0)}(\mathbf{W})\right)\right) \quad (3.29)$$

$\mathbf{P}_j(\mathbf{W})$ lautet:

$$\mathbf{P}_j(\mathbf{W}) = \mathbf{I} + \alpha_{jj}\Delta t\mathbf{M}^{-1}\frac{d\mathbf{R}}{d\mathbf{W}}(\mathbf{W}^{\mathbf{T}_n} + \alpha_{j,j-1}\Delta tk_{j-1}(\mathbf{W})). \quad (3.30)$$

Der folgende Algorithmus verwendet die Formulierung des impliziten Runge-Kutta-Verfahrens (3.25), das die innere Iteration des Newtonschen Verfahrens nach einem Schritt abbricht:

$$\begin{aligned} k_1(\mathbf{W}) &:= -[\mathbf{P}_1(\mathbf{W})]^{-1}\mathbf{M}^{-1}\mathbf{R}(\mathbf{W}^{\mathbf{T}_n}) \\ k_2(\mathbf{W}) &:= -[\mathbf{P}_2(\mathbf{W})]^{-1}\mathbf{M}^{-1}\mathbf{R}(\mathbf{W}^{\mathbf{T}_n} + \alpha_{21}\Delta tk_1(\mathbf{W})) \\ &\vdots \\ k_s(\mathbf{W}) &:= -[\mathbf{P}_s(\mathbf{W})]^{-1}\mathbf{M}^{-1}\mathbf{R}(\mathbf{W}^{\mathbf{T}_n} + \alpha_{s,s-1}\Delta tk_{s-1}(\mathbf{W})) \end{aligned} \quad (3.31)$$

Wobei die konservativen Variablen für den nächsten Zeitschritt durch

$$\mathbf{W}^{\mathbf{T}_{n+1}} = \mathbf{W}^{\mathbf{T}_n} + \alpha_{s+1,s}\Delta tk_s(\mathbf{W}). \quad (3.32)$$

definiert werden. Wenn die Aktualisierung als $\mathbf{W}^{(0)} := \mathbf{W}^{\mathbf{T}_n}$ definiert wird, und wenn an $k_1(\mathbf{W})$ und $k_2(\mathbf{W})$ aus (3.31) berücksichtigt werden, kommt die folgende Rekursionsformel

$$\mathbf{W}^{(j)} = \mathbf{W}^{\mathbf{T}_n} + \alpha_{j+1,j}\Delta t[\mathbf{P}_j(\mathbf{W})]^{-1}\mathbf{M}^{-1}\mathbf{R}(\mathbf{W}^{(j-1)}) \quad (3.33)$$

heraus, die den nächsten Zeitschritt darstellt. Nun kann die Formel (3.31) induktiv wieder formuliert werden

$$\begin{aligned}
\mathbf{W}^{(0)} &:= \mathbf{W}^{\text{T}_n} \\
\mathbf{W}^{(1)} &= \mathbf{W}^{(0)} - \alpha_{21} \Delta t \mathbf{P}_1(\mathbf{W})^{-1} \mathbf{M}^{-1} \mathbf{R}(\mathbf{W}^{(0)}) \\
&\vdots \\
\mathbf{W}^{(s)} &:= \mathbf{W}^{(0)} - \alpha_{s+1,s} \Delta t \mathbf{P}_s(\mathbf{W})^{-1} \mathbf{M}^{-1} \mathbf{R}(\mathbf{W}^{(s-1)}) \\
\mathbf{W}^{\text{T}_{n+1}} &= \mathbf{W}^{(s)}.
\end{aligned} \tag{3.34}$$

In Algorithmus (3.34), wenn ein stufige Schema beispielweise CFL-Zahl = ∞ und exakte Residuumsfunktion \mathbf{R} in der Algorithmus 3.34 berücksichtigt wird, ist die Formulierung:

$$\mathbf{W}^{n+1} = \mathbf{W}^n - \left(\frac{d\mathbf{R}(\mathbf{W}^{(n)})}{d\mathbf{W}} \right)^{-1} \mathbf{R}(\mathbf{W}^{(n)}). \tag{3.35}$$

Die ist nicht anderes als Newton-Verfahren. Somit kann das interpretiert werden, dass der Algorithmus (3.34) implizit Runge-Kutta Verfahren durch direkten Anwendung der Newton-Verfahren gelöst wird.

Die Formulierungen (3.34) zeigen, dass die lineare Gleichung

$$\mathbf{P}_j(\mathbf{W}) \mathbf{h}_j = \alpha_{j+1,j} \Delta t \mathbf{M}^{-1} \mathbf{R}(\mathbf{W}^{(j-1)}) \tag{3.36}$$

für jede Stufe gelöst werden muss. \mathbf{P}_j aus (3.30) wird in die Gleichung (3.36) eingesetzt und wie folgt umgestellt

$$\left((\Delta t)^{-1} \mathbf{M} + \alpha_{jj} \frac{d\mathbf{R}(\mathbf{W}^{(j-1)})}{d\mathbf{W}} \right) \mathbf{h}_j = \alpha_{j+1,j} \mathbf{R}(\mathbf{W}^{j-1}). \tag{3.37}$$

Jedes Element der vektorwertigen Funktion \mathbf{R}_i ist abhängig von einem unbekannten \mathbf{W}_i , das im Kontrollvolumen i liegt, sowie von den direkten Nachbarn $\mathbf{W}_{j,j \in \mathcal{N}(i)}$ und deren jeweiligen Nachbarn $\mathbf{W}_{k,k \in \mathcal{N}(j), j \in \mathcal{N}(i)}$. Beispielsweise beträgt die Anzahl der Abhängigkeiten in einem 3D-Hexaedergitter:

$$|k, k \in \mathcal{N}(j), j \in \mathcal{N}(i)| = 25. \tag{3.38}$$

Deshalb hat jede Zeile der Matrix $\frac{d\mathbf{R}}{d\mathbf{W}}$ 25 Elemente ungleich null. Das heißt, die Zahl der Elemente eines Gitters mit d Freiheitsgraden und N Punkten ist $N \cdot d \cdot d \cdot 25$. Aus der Grundlagengleichung (2.1) wird klar, dass $d = 5$ sein muss. Bei Verwendung der Programmiersprache C werden 8 Bytes verbraucht. Somit liegt der gesamte Speicherbedarf einer Matrix bei:

$$25 \cdot 25 \cdot 8 \cdot N \text{ Bytes} = 5000 N \text{ Bytes}. \tag{3.39}$$

Für eine Matrix $\frac{d\mathbf{R}}{d\mathbf{W}}$ mit $N = 1e06$ Gitterzellen werden beispielsweise etwa 5GB verbraucht. Obwohl moderne Hardware-Cluster einen höheren Speicherbedarf decken, ist dieser nicht für eine Lösung einer

Fullmatrix mit einem direkten Löser, der auf Dekomposition basiert, anwendbar. Die Verwendung einer numerischen Ableitung erster Ordnung hat den Vorteil, dass der Stern am Punkt i nur die Information der nächsten Nachbarn benötigt.

Die dazugehörige Jakobimatrix $\frac{\partial \mathbf{R}^{1st}}{\partial \mathbf{W}}$ ist viel weniger speicherintensiv als exakte Derivat $\frac{\partial \mathbf{R}}{\partial \mathbf{W}}$. Im Vergleich zu (3.38) ist die Zahl der Nachbarn der 3D-Hexaedernetze 6. Daher ist die Anzahl der Einträge der $\frac{\partial \mathbf{R}^{1st}}{\partial \mathbf{W}}$ 7. Die heißt, das Preconditioning Matrix **Prec** erfordert $\frac{25}{7} \approx 3,57$ Mal weniger Speicherplatz als die exakte Ableitung $\frac{\partial \mathbf{R}}{\partial \mathbf{W}}$.

Der Operator $\|\mathbf{A}_{ij}^{Roe}\|$, in der konvektiven Term der diskretisierten Gleichungssystem ist, und die effektiven Viskositäten μ_{eff} und κ_{eff} werden lokal als konstant angenommen. Das heißt, dass die Konstruktion des Tensors von dritter Ordnung vermieden wird. Beispielsweise wird die erster Ordnung Diskretisierung von Matrix Dissipation zweiter Ordnung oder Roe Schema zweiter Ordnung vorgestellt. Die Diskretisierung erster Ordnung des nichtviskosen Terms aus Gleichung (2.1) kann dargestellt werden durch:

$$\int_{\partial\Omega_i} \mathbf{f}_c \cdot \mathbf{n} ds \approx \sum_{j \in N(i)} \frac{1}{2} [(\mathbf{f}_c \cdot \mathbf{n}_{ij})(\mathbf{W}_i) + (\mathbf{f}_c \cdot \mathbf{n}_{ij})(\mathbf{W}_j)] - \frac{1}{2} |\mathbf{A}_{ij}^{Roe}| (\mathbf{W}_j - \mathbf{W}_i). \quad (3.40)$$

Die Approximation der Ableitung des konvektiven Fluss erster Ordnung ist:

$$\frac{\partial \mathbf{R}_i^{1st,conv}}{\partial W_k} = \begin{cases} \sum_{j \in N(i)} |\mathbf{A}_{ij}^{Roe}| & k = i \\ -|\mathbf{A}_{ik}^{Roe}| & k \in N(i) \\ 0 & k \neq i, k \notin N_i \end{cases} \quad (3.41)$$

Beispielsweise sieht eine Ableitung erster Ordnung in der i -ten Reihe an den Nachbarnpunkten j_1, \dots, j_6 wie folgt aus:

$$\begin{aligned} \frac{\partial \mathbf{R}_i^{1st,conv}}{\partial \mathbf{W}} = & \left(0, \dots, 0, \frac{\partial \mathbf{R}_i^{1st,conv}}{\partial W_{j_1}}, 0, \dots, 0, \frac{\partial \mathbf{R}_i^{1st,conv}}{\partial W_{j_2}}, 0, \dots, 0, \frac{\partial \mathbf{R}_i^{1st,conv}}{\partial W_{j_3}}, \right. \\ & \left. 0, \dots, 0, \frac{\partial \mathbf{R}_i^{1st,conv}}{\partial W_{j_4}}, 0, \dots, 0, \frac{\partial \mathbf{R}_i^{1st,conv}}{\partial W_{j_5}}, 0, \dots, 0, \frac{\partial \mathbf{R}_i^{1st,conv}}{\partial W_{j_6}} \right) \end{aligned} \quad (3.42)$$

Die Konstruktion der viskosen Ableitung $\frac{\partial \mathbf{R}_i^{1st,visc}}{\partial W_k}$ kann in Artikel [9] nachvollgezogen werden. Der Preconditioner muss schnell ausgeführt werden, deswegen wird das Compressed Sparse Row Format (CSR) verwendet, um die Matrix und das zugehörige lineare Gleichungssystem zu speichern. Details zu diesem Format sind in [18] zu finden. In diesem Versuch wird nur der stationäre Strömungsfall berücksichtigt. Daher ist der Zeitschritt folgendermaßen gegeben:

$$\Delta t_i := CFL \cdot \text{vol}(\Omega_i) \cdot \left[\sum_{j \in N(i)} \frac{1}{2} (|V_{ij}| + a_{ij} \text{vol}(\Omega_{ij})) + \frac{8(\mu_{eff})_{ij} s \text{vol}(\Omega_{ij})}{\|x_{p_i} - x_{p_j}\|_{2\rho_i}} \left(\max \left\{ \frac{3}{4}, \frac{(\kappa_{eff})_{ij}(\gamma - 1)}{(\mu_{eff})_{ij}} \right\} \right) \right]^{-1} \quad (3.43)$$

Hierbei ist $C_v := 8$ und Δt wird im linearen Gleichungssystem (3.37) für ΔT eingesetzt,

$$\Delta T := \text{diag}(\text{diag}(\Delta t_i)) \in \mathbb{R}^{5N \times 5N}. \quad (3.44)$$

Die Matrix \mathbf{M} umfasst für turbulente Strömungen fünf Variablen. In der Reynolds-gemittelten Strömung reduziert sich dies lediglich auf ν . Daher wird (3.44) umgeschrieben zu:

$$\Delta T := \text{diag}(\Delta t_i) \in \mathbb{R}^{N \times N}. \quad (3.45)$$

Hier wird der Relaxationsparameter ϵ dem linearen Gleichungssystem (3.37) hinzugefügt und nach Anwendung der Approximation der Jakobimatrix und Zeitschrittmatrix ΔT sieht das lineare Gleichungssystem unter direkter Anwendung des Newton-Verfahrens wie folgt aus:

$$\left((\Delta T)^{-1} \mathbf{M} + \epsilon \alpha_{jj} \frac{d\mathbf{R}^{1st}(\mathbf{W}^{(j-1)})}{d\mathbf{W}} \right) \mathbf{h}_j = \alpha_{j+1,j} \mathbf{R}(\mathbf{W}^{(j-1)}). \quad (3.46)$$

3.3 Levenberg-Marquardt Algorithmus

Wie oben in Kapitel 3.2 gezeigt, die implizite Runge-Kutta Verfahren (3.34) durch direkten Anwendung der Newton-Verfahren gelöst wird. Die Newton-Verfahren hat einfache Konstruktion und dadurch ist die Berechnen günstig und konvergiert die Lösung schnell. Jedoch ist schwierig die konvergierte Lösung zu erhalten wenn die Startnäherung sehr weit von der richtigen Lösung ist. Der Anlass lässt sich andere Strategie verwenden um der Glätter zu stabilisieren. Zur Verbessern der Robustheit der Newton-Verfahren wird Levenberg-Marquardt Algorithmus verwendet. Nun versuchen wir statt des linearen Gleichungssystems (3.46) eine andere lineare Gleichungssystem durch Levenberg-Marquardt Algorithmus aufzubauen. Nun versuchen wir der Lösung der linearen Gleichungssystems (3.46) durch der kleinsten Quadraten Methode zu optimieren. Als Anfang denken wir an L2-Norm der lineare Gleichungssystem (3.46),

$$\left\| \epsilon \alpha_{jj} \frac{d\mathbf{R}^{1st}(\mathbf{W}^{(j-1)})}{d\mathbf{W}} \mathbf{h}_j - \alpha_{j+1,j} \mathbf{R}(\mathbf{W}^{(j-1)}) \right\|_2^2 + \gamma_j \|\mathbf{h}_j\|_2^2, \quad (3.47)$$

wobei γ_j einen Regularisierungsparameter γ_j [?] mit $\gamma_j \geq 0$ und

$$\gamma_j = [(\Delta T)^{-1} \mathbf{M}]^2. \quad (3.48)$$

Zur Erhalten der optimierten Lösung durch der kleinsten Quadraten Methode wird die Lösung gesucht, die die L2-Norm minimiert (3.47) wird. Das heißt, es soll das Minimierungsproblem

$$\left\| \epsilon \alpha_{jj} \frac{d\mathbf{R}^{1st}(\mathbf{W}^{(j-1)})}{d\mathbf{W}} \mathbf{h}_j - \alpha_{j+1,j} \mathbf{R}(\mathbf{W}^{(j-1)}) \right\|_2^2 + \gamma_j \|\mathbf{h}_j\|_2^2 = \min_{\mathbf{h}_j}! \quad (3.49a)$$

gelöst werden. Das Minimierungsproblem (3.49a) ist äquivalent zu

$$\left\| \begin{pmatrix} \epsilon \alpha_{jj} \frac{d\mathbf{R}^{1st}(\mathbf{W}^{(j-1)})}{d\mathbf{W}} \\ \sqrt{\gamma_j} \mathbf{I} \end{pmatrix} \mathbf{h}_j - \begin{pmatrix} \alpha_{j+1,j} \mathbf{R}(\mathbf{W}^{(j-1)}) \\ 0 \end{pmatrix} \right\|_2^2 = \min_{\mathbf{h}_j}!. \quad (3.50)$$

Die Lösung der Minimierungsproblem (3.50) kann durch Lösen der Normalengleichung

$$\left(\gamma_j \mathbf{I} + \epsilon \alpha_{jj} \left(\frac{d\mathbf{R}^{1st}(\mathbf{W}^{(j-1)})}{d\mathbf{W}} \right)^T \frac{d\mathbf{R}^{1st}(\mathbf{W}^{(j-1)})}{d\mathbf{W}} \right) \mathbf{h}_j = \alpha_{j+1,j} \left(\frac{d\mathbf{R}^{1st}(\mathbf{W}^{(j-1)})}{d\mathbf{W}} \right)^T \mathbf{R}(\mathbf{W}^{(j-1)}) \quad (3.51)$$

bestimmt werden. Die Lösung der Normalengleichung (3.51) ist dieselbe, die Lösung durch Methode der kleinsten Quadrate ist. Der Levenberg-Marquardt Verfahren gewinnt die Ungleichheit

$$\begin{aligned} \gamma_j \|\mathbf{h}_j\|_2^2 &\leq \left\| \epsilon \alpha_{jj} \frac{d\mathbf{R}^{1st}(\mathbf{W}^{(j-1)})}{d\mathbf{W}} \mathbf{h}_j - \alpha_{j+1,j} \mathbf{R}(\mathbf{W}^{(j-1)}) \right\|_2^2 + \gamma_j \|\mathbf{h}_j\|_2^2 \\ &= \min_{\mathbf{h}_j} \left[\left\| \epsilon \alpha_{jj} \frac{d\mathbf{R}^{1st}(\mathbf{W}^{(j-1)})}{d\mathbf{W}} \mathbf{h}_j - \mathbf{R}(\mathbf{W}^{(j-1)}) \right\|_2^2 + \gamma_j \|\mathbf{h}_j\|_2^2 \right] \\ &\leq \|\mathbf{R}(\mathbf{W}^{(j-1)})\|_2^2, \end{aligned}$$

Letztendlich ist die Ungleichung gegeben durch:

$$\|\mathbf{h}_j\|_2^2 \leq \frac{\|\mathbf{R}(\mathbf{W}^{(j-1)})\|_2^2}{\sqrt{\gamma_j}}. \quad (3.52)$$

Somit ist die Regularisierungsparameter γ_n ein Dämpfung der Lösung \mathbf{h}_j . Hier kann das eindeutig erfolgen, dass das Levenberg-Marquardt Verfahren für großem γ konvergiert werden kann. Andere Seite mag es langsam konvergiert werden, when γ groß und die Lösung \mathbf{h}_j klein ist. Nun hat γ die Rolle de Zeitschrittwahl und damit $\gamma_j \mathbf{I}$ spielt die Rolle der Stabilisierung der Konvergierung. Unter Kenntnis darf es verstehen, dass die CFL-Zahl in einem Punkt verändern soll. Das heißt, dass generell in initiale Pahase der unsere Lösungsschema kleine CFL-Zahl ausgewählt wird. Nach einem guten Initialschätzung, die eine größere CFL-Zahl findet, mag die Konvergenzgeschwindigkeit beschleunigt zu werden. Jedoch ist es generell unmöglich zu beurteilen wo und wann die Iteration in der Nähe von Lösung ist. Außerdem wird einige Dämpfung dafür verbraucht. Die Regularisierungsparameter γ_n hat einige Effekte, die sich in Artikel[19] bezieht:

1. Für alle $\gamma_j > 0$ ist Koeffizient Matrix positiv definiert und \mathbf{h}_j stieg ab.
2. Für große γ_j folgt

$$\mathbf{h}_j \simeq -\frac{1}{\gamma_j} \frac{d\mathbf{R}^{1st}(\mathbf{W}^{(j-1)})}{d\mathbf{W}} \mathbf{R}(\mathbf{W}^{(j-1)}) \quad (3.53)$$

Das heißt, das ist ein kleine Schritt in die steileste Abstiegrichtung und geeignet wenn aktuelle Iteration weit von der Lösung ist.

3. Wenn γ_j sehr klein ist, dann das Wert \mathbf{h} mit Levenberg-Marquardt Verfahren ist ungefähr gleich wie das mit Gauss-Newton Verfahren, welche geeignet für Finale Phase der Iteration ist, wann \mathbf{W} sehr in der Nähe von \mathbf{W}^* ist. Das heißt, wir können die finale Kovergenz $\mathbf{R}(\mathbf{W}^*)$ erhalten.

Die regularisierungsparameter γ wird in diesem Arbeit durch Veränderung der CFL-Zahl in jedes Iteration und durch (3.48) aktuarisiert. Die Regel der Veränderung der CFL-Zahl ist in (4.2) gegeben.

3.4 Lösungsmethode für das linear Gleichungssystem

In diesen Teil wird die Lösungsmethode zum approximativen Lösen der linearen Gleichungssystem dargestellt. Die Lösungsmethode dieses Arbeits muss die linear-Gleichungssystem, welche die mit direkten Anwendung der Newton-Verfahren (3.46) oder mit Anwendung der Levenberg-Marquardt Algorithmus (3.51) ist, in jeden Runge-Kutta Stufen approximativ gelöst werden. Die Berechnen der linear-Gleichungssystemen in jeden Iteration muss aber nicht maximale Geuigkeit sondern soll sie effizient sein.

Hier werden (Block) Jacobi-Verfahren und (Block)(Symmetrisch) Gauss-Seidel Verfahren vorgestellt.

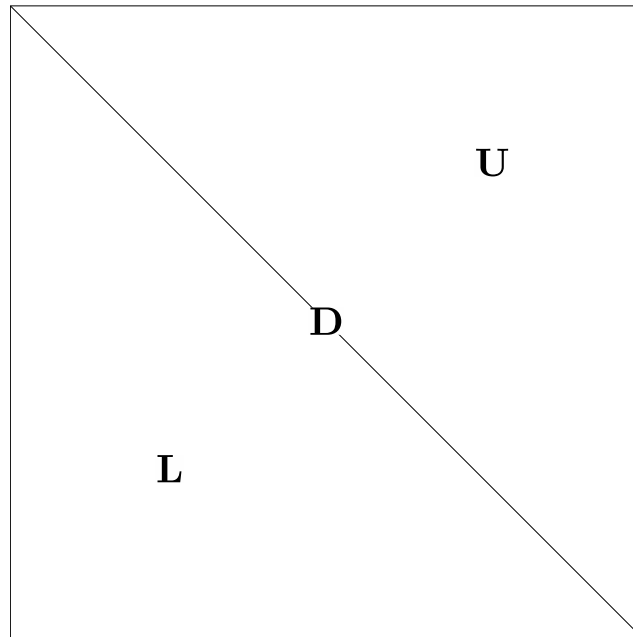


Abb. 3.4: Ein Skizze der Dekomposition einer quadratischen Matrix

Stationär iterativ Löser verwendet klassisch die Dekomposition der Matrix \mathbf{P} , wenn ein linear Gleichungssystem $\mathbf{P} \cdot \mathbf{h} = \mathbf{b}$ betrachtet wird, wobei ein Vektor \mathbf{h} eine Lösung des linearen Gleichungssystems, \mathbf{P} ein $n \times n$ quadratische Matrix und n -te rechtsseitig Vector \mathbf{b} ist. Residuum ist dann $\mathbf{b} - \mathbf{P}\mathbf{h}$. Die Formulierung fängt die Dekomposition einer quadratische Matrix \mathbf{P} an,

$$\mathbf{P} = \mathbf{D} + \mathbf{U} + \mathbf{L} \quad (3.54)$$

Wobei \mathbf{L} untere Dreiecksmatrix, \mathbf{D} Diagonalmatrix und \mathbf{U} obere Dreiecksmatrix sind, wie im Bild 3.4 gezeigt. Die Jacobi-Iteration versucht die i -te Komponente nächsten Approximation zu detektieren. Daher wird i -te Komponente der Residuumsvektor ausgelöst,

$$(\mathbf{b} - \mathbf{P}\mathbf{h}_{k+1})_i = 0, \quad i = 1, \dots, n. \quad (3.55)$$

Finalform der Jacobi-Verfahren als komponente Darstellung sieht

$$\mathbf{h}_i^{(k+1)} = (\mathbf{P}_{i,i})^{-1} \left(\mathbf{b}_i - \sum_{j=i, j \neq i} \mathbf{P}_{i,j} \mathbf{h}_j^{(k+1)} \right), \quad i = 1, \dots, n \quad (3.56)$$

aus. In ähnlicher Weise detektiert die Gauss-Seidel Iteration die i -te Komponente der aktuellen Lösung in der Ordnung $i = 1, 2, \dots, n$. Hier wird die i -te Komponente der Residualvektor ausgelöst. Die Gauss-Seidel Verfahren wird aber die approximierte Lösung sofort aktualisiert, nachdem ein neue Komponente berechnet wird. Das heißt, die neue Komponente $\mathbf{h}_i^{(k)}$, $i = 1, 2, \dots, n$ kann mit der Vektor, welche in jedes Relaxation Schritt wieder definiert wird, verändert werden. Bei $i = 1, 2, \dots, n$ ist die Lösung der Gauss-Seidel-Verfahren,

$$\mathbf{h}_i^{(k+1)} = (\mathbf{P}_{i,i})^{-1} \left(\mathbf{b}_i - \sum_{j=i}^{i-1} \mathbf{P}_{i,j} \mathbf{h}_j^{(k+1)} - \sum_{j=i+1}^N \mathbf{P}_{i,j} \mathbf{h}_j^{(k)} \right), \quad i = 1, \dots, n. \quad (3.57)$$

Wobei ist $\mathbf{P} = (\mathbf{P}_{ij})_{1 \leq i, j \leq N}$ und \mathbf{b} ist rechterseits von der linearen System (3.46) oder (3.51). Initiale Schätzung $\mathbf{h}^{(0)}$ ist 0. Symmetrisch Gauss-Seidel Verfahren führt ein vorwärts Sweep nachdem ein rückwärts Sweep durch.

In Teil 4 wird die numerische Testfällen zum nährungsweisen Lösen des Linearesystems (3.46) 5 und des Linearesystems (3.51) 5 bis 25 (50) symmetrische Sweeps von (3.57) getestet. Ein Sweep hier bedeutet ein vor- und rückwärts Sweep.

4 Numerische Beispiele

Im vorangegangenen Abschnitt 3 wurde die Lösungsmethode zur Lösen der gewöhnliche diskretisierte Differentialgleichungen (2.48) durch FAS Mehrgitter Verfahren, die als Glätter die mehrstufige implizit Runge-Kutta Verfahren verwendet wird, vorgestellt. Newton-Verfahren wird hier zur Lösen der nicht-Lineare Gleichungssystem verwendet um die Stufe-Koeffizienten zu bestimmen. Als äquivalente Idee der Newton-Verfahren wird Levenberg-Marquardt Algorithmus vorgestellt, welche robstere Variante der Newton-Verfahren ist. In diesem Teil sollen nun drei numerische Beispiele zum Testen der neuen strategie vorgestellt werden.

Hierfür werden die zwei lineare-Gleichungssystem, die jede Stufe der impliziten Runge-Kutta Verfahren approximativ gelöst werden muss, welche für Newton-Verfahren (3.46) und für Levenberg-Marquardt Algorithmus (3.51) ist, in allen Testfällen implementiert und getestet. Zur Vergleichen der Robustheit zwischen Newton-Verfahren und Levenberg-Marquardt Algorithmus werden die Berechnungen in 2-D Testfällen mit Parallelisierung und ohne Parallelisierung durchführt.

Im ersten Testfall wird das Profil NACA0012 als 2D-Geometrie in einer Euler-Strömung getestet, in nächsten Testfall in einer laminaren Strömung. In beiden Testfällen wird ein strukturiertes Gitter eingesetzt und die grob, mittel und fein Gitterstufe betrachtet. In den letzten Testfall hingegen wird ein dreidimensionaler Körper, ein Zylinder mit halbkugelförmigen (hemisphärischen) Enden getestet. Sowohl die Geometrie als auch die Randbedingungen werden ausgewählt wie in [20]. Außerdem wird für diesen Testfall ein tetrahedralische struktrierten Gitter ausgewählt und auch die grob, mittel und fein Gitterstufen betrachtet. In der Arbeit wird nur in den letzten Testfall turbulente Strömung betrachtet und mit einen Verhältnis 1:3 Schritte beim Iteration zwischen mittleren Strömungs- und turbulenten Strömungsgleichung durchführt. Das Residuum wird von Volumen gewichtete Norm

$$\begin{aligned}
 \text{Dichte Residuum}(n) &:= \frac{\sqrt{\sum_{j=1}^N \frac{(\mathbf{R}_{j,\text{mean},\rho}(\mathbf{W}^{\text{T}n}))^2}{(\text{vol}(\Omega_j))^2}}}{\sqrt{\sum_{j=1}^N \frac{(\mathbf{R}_{j,\text{mean},\rho}(\mathbf{W}_{\infty}))^2}{(\text{vol}(\Omega_j))^2}}}, \\
 \text{Turbulenz Residuum}(n) &:= \frac{\sqrt{\sum_{j=1}^N \frac{(\mathbf{R}_{j,\text{turb},\rho}(\tilde{\nu}^{\text{T}n}))^2}{(\text{vol}(\Omega_j))^2}}}{\sqrt{\sum_{j=1}^N \frac{(\mathbf{R}_{j,\text{turb},\rho}(\tilde{\nu}_{\infty}))^2}{(\text{vol}(\Omega_j))^2}}}
 \end{aligned} \tag{4.1}$$

kalikuliert, das in Bezug auf die mit freien Strömungswerte bewerteten Residuum normalisiert. Die Berechnungen werden gestoppt, sobald das Dichter residuum die 14. Ordnung der Magnitude erreicht.

Alle Testfälle mit Levenberg-Marquardt Algorithmus haben erfolgreich konvergiert, nur wenn ohne Parallelisierung ist. Das heißt, Parallelisierung mit Levenberg-Marquardt Algorithmus sind nicht erfolgreich außer laminar Testfälle. Somit sind alle Testfälle mit Levenberg-Marquardt Algorithmus ohne Parallelisierung.

CFL-Cahl verändert sich in Iterationsschritt, da in Initial Phase kleine und Finale große *CFL*-Zahl

genommen werden soll, ist eine gewöhnlichen Technik um die Lösung zu konvergieren. Bis zehnte Iterationen bleibt die initiale CFL -Zahl. Danach steigt sie sich mit einem Anstiegfaktor α bis Finale CFL -Zahl CFL_{final} an. Das wird als Formel,

$$\begin{aligned} \mathbf{CFL} &= \min\{\mathbf{CFL}_{\text{init}} \cdot f(n), \mathbf{CFL}_{\text{max}}\}, \\ f(n) &= \begin{cases} 1, & n < 10, \\ \alpha^{n-10} & n \geq 10, \end{cases} \quad , \alpha > 0. \end{aligned} \quad (4.2)$$

gezeigt. Als Zusammenfassung der gezeigten Lösungsmethode in Kapitel 3.4 kommt die folgende Aufgliederung:

1. Drei Stufen des Runge-Kutta Verfahren (3.25) wird mit Stufe-Koeffizienten $\alpha_{21} = 0,15, \alpha_{32} = 0,4, \alpha_{43} = 1$, und $\alpha_{11} = \alpha_{22} = \alpha_{33} = 1$ verwendet.
2. Newton-Verfahren wird zum Lösen des Linearsystem (3.46) und Levenberg-Marquardt Algorithmus wird zum Lösen des Linearsystem (3.51) verwendet, welche dazugehöriges lineare Gleichungssystem von der Gleichung (3.57) gelöst wird.

4.1 2D-Testfälle

In den Testfällen (a) und (b) wird das Profil NACA0012 getestet. Der klassische Testfall lässt sich in der Literatur finden[21]. In beiden Testfällen wird mit dem nichtlinearen Mehrgitterzyklus "4w" gerechnet. Als Abbruchkriterium wird $1e-14$ gewählt. Außerdem werden für die beiden Fälle drei verschiedene Gittergrößen getestet. In den Abbildung 4.1 sind exemplarisch das Profil und das Gitter dargestellt. Weiter Details können im Artikel [21] gefunden werden. die Zusammenfassung der Testparametern für die 2-D Testfälle sind,

- (a) Profil NACA0012: Berechnet mit der Eulerschen Gleichung und den Parametern: $Re = 5.000$, $Ma = 0,5$, $AoA = 0,0^\circ$.
- (b) Profil NACA0012: Berechnet mit der Eulerschen Gleichung mit viskosem Term (laminare Strömung) und den Parametern: $Re = 5.000$, $Ma = 0,5$, $AoA = 0,0^\circ$.

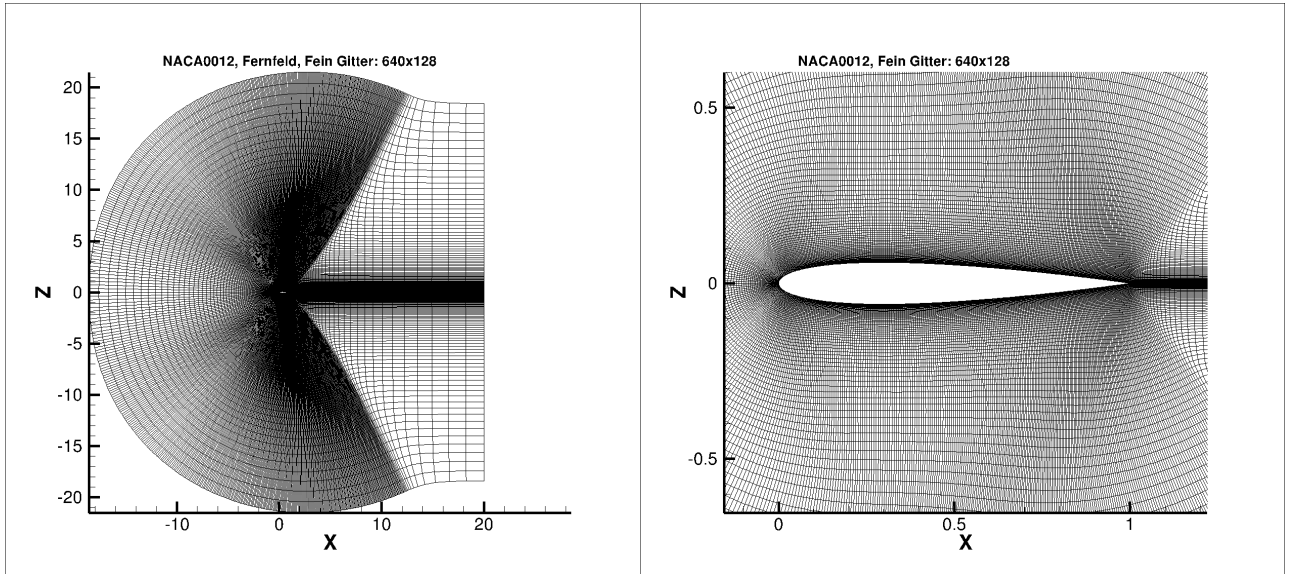


Abb. 4.1: Gitter des Profils NACA0012, Gittergröße: 512×256

4.1.1 Testfall(a):NACA0012, Euler-Strömung

In diesem Testfall werden die Lösungen durch Newton-Verfahren ohne Parallelisierung und mit Parallelisierung konvergiert, während die Lösungen durch Levenberg-Marquardt Algorithmus nur ohne Parallelisierung konvergiert werden. In der Tabelle (4.2) sind jeweils für die eulere Strömung mit der Newton-Verfahren und in Tabelle (4.3) eulere Strömung mit dem Levenberg-Marquardt Verfahren die Parameter für den Algorithmus aufgelistet.

Parameter	Wert		
	Grob	Mittel	Fein
Gittergröße	160×32	320×64	640×128
Zahl der Zelle	15712	62144	247168
Zahl der Punkten	21888	84736	333312
Zahl der Domänen für Parallelisierung	3	4	6

Tab. 4.1: Gitter Data von NACA0012 auf dem Eulerströmung

Parameter	Wert		
	Grob	Mittel	Fein
Runge-Kutta-Stufe	3	3	3
Initiale CFL-Zahl	10	10	10
Finale CFL-Zahl	1000	1000	1000
Relaxationsparameter	1,0	1,0	1,0
Anstiegfaktor für CFL-Zahl	5	5	5
Zahl der Gauss-Seidel Sweeps	5	5	5
Mehrgitterzyklus	4W	4W	4W

Tab. 4.2: Algorithmischer Parameter durch Newton-Verfahren auf NACA0012 in Eulerströmung

Parameter	Wert		
	Grob	Mittel	Fein
Runge-Kutta-Stufe	3	3	3
Initiale CFL-Zahl	10	10	10
Finale CFL-Zahl	1000	1000	1000
Relaxationsparameter	0,6	0,6	0,6
Anstiegfaktor für CFL-Zahl	5	5	5
Zahl der Gauss-Seidel Sweeps	5	5	5
Mehrgitterzyklus	4W	4W	4W

Tab. 4.3: Algorithmischer Parameter für Parallelisierung Test durch Levenberg-Marquardt Algorithmus auf NACA0012 Eulerströmung

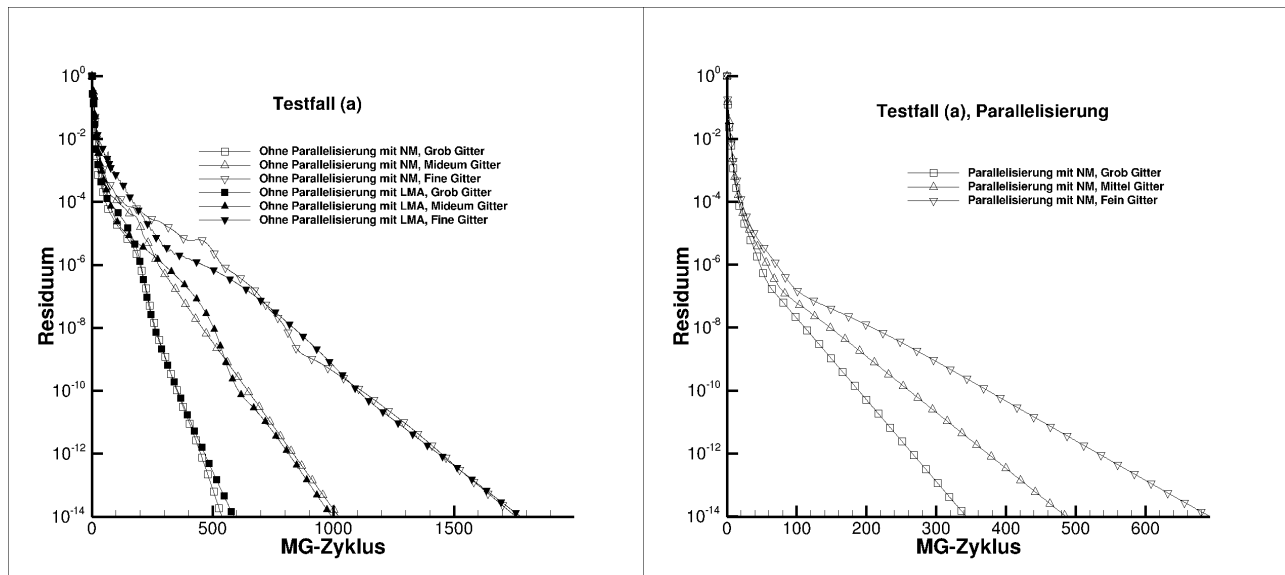


Abb. 4.2: Konvergenzhistroy von euler Strömung, ohne Parallelisierung (Links), mit Parallelisierung (Rechts)

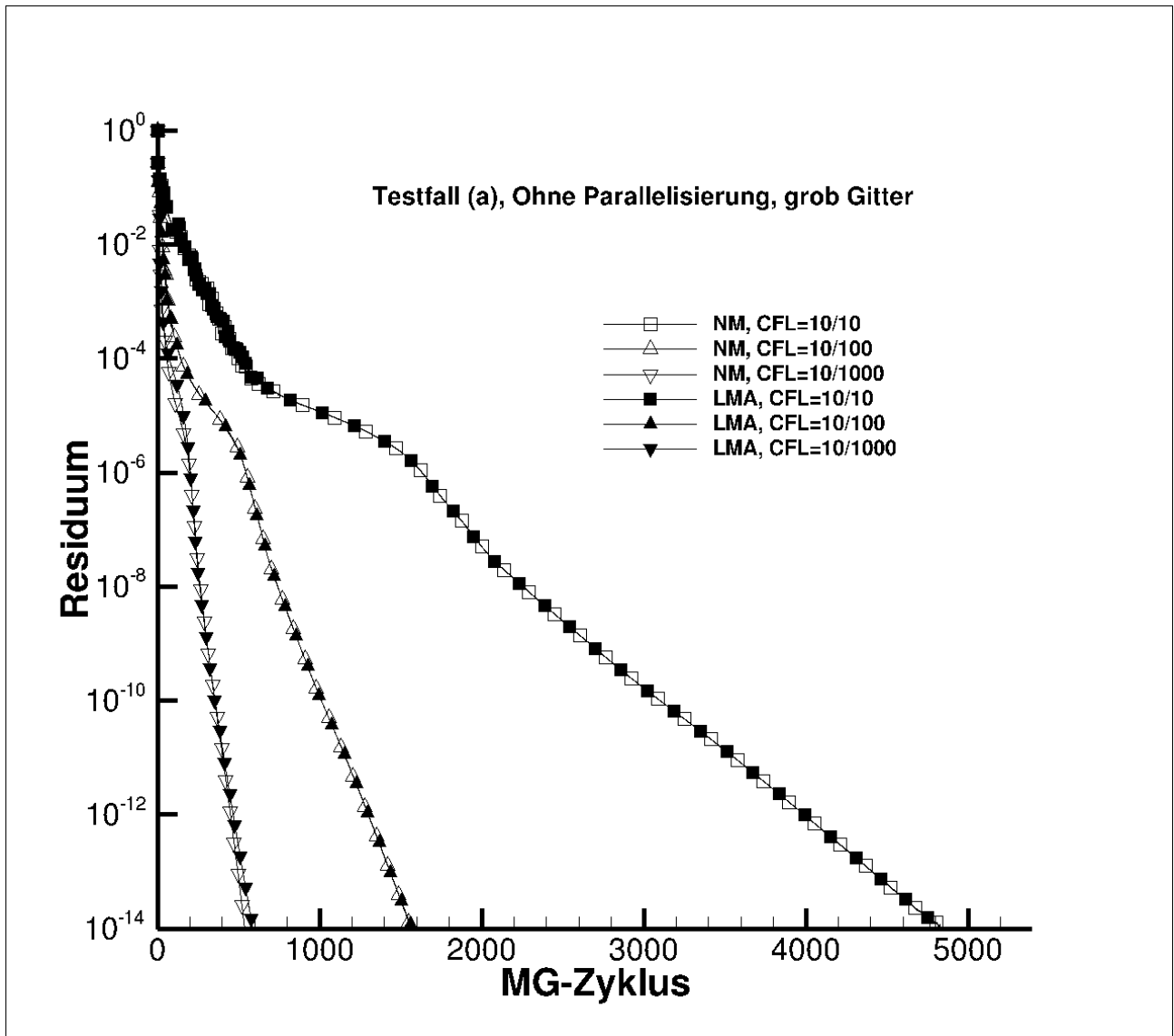


Abb. 4.3: Konvergenzhistorie ohne Parallelisierung Testfälle in Vergleich zu verschiedene CFL-Zahlen auf dem grob Gitter, Eulere Strömung

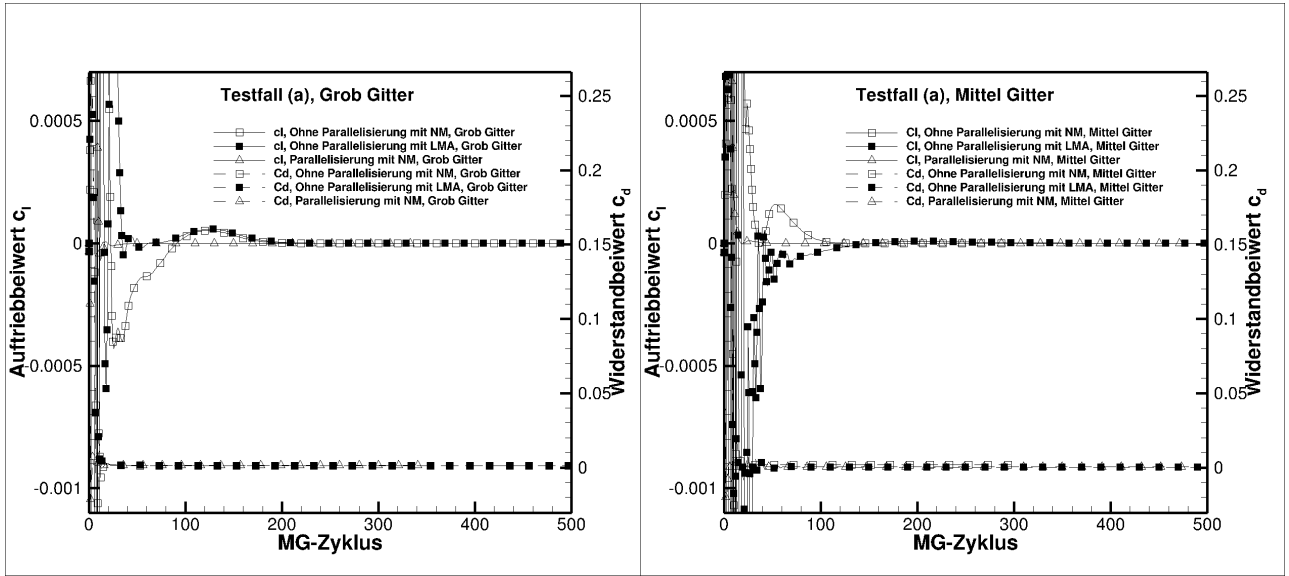


Abb. 4.4: Vergleich der Konvergenzhistorie von Wider-

Abb. 4.5: Vergleich der Konvergenzhistorie von Auftrieb-

standsbeiwert, Eulere Strömung, grob Gitter

beiwert, Eulere Strömung, mittel Gitter

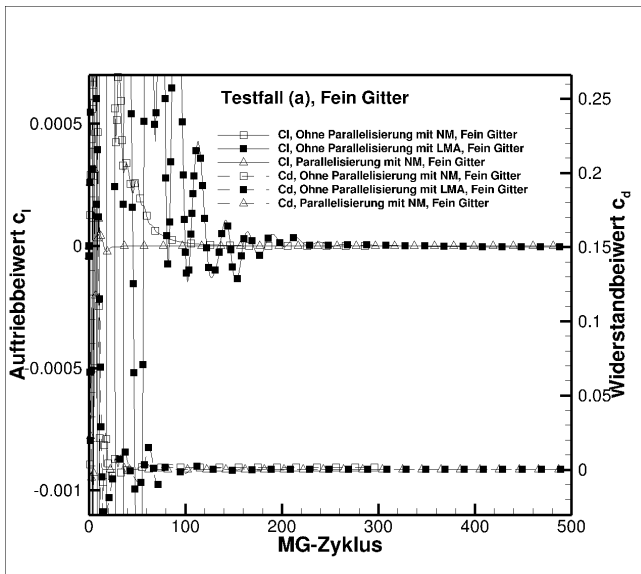


Abb. 4.6: Vergleich der Konvergenzhistorie von Auftrieb-

beiwert, Eulere Strömung, fein Gitter

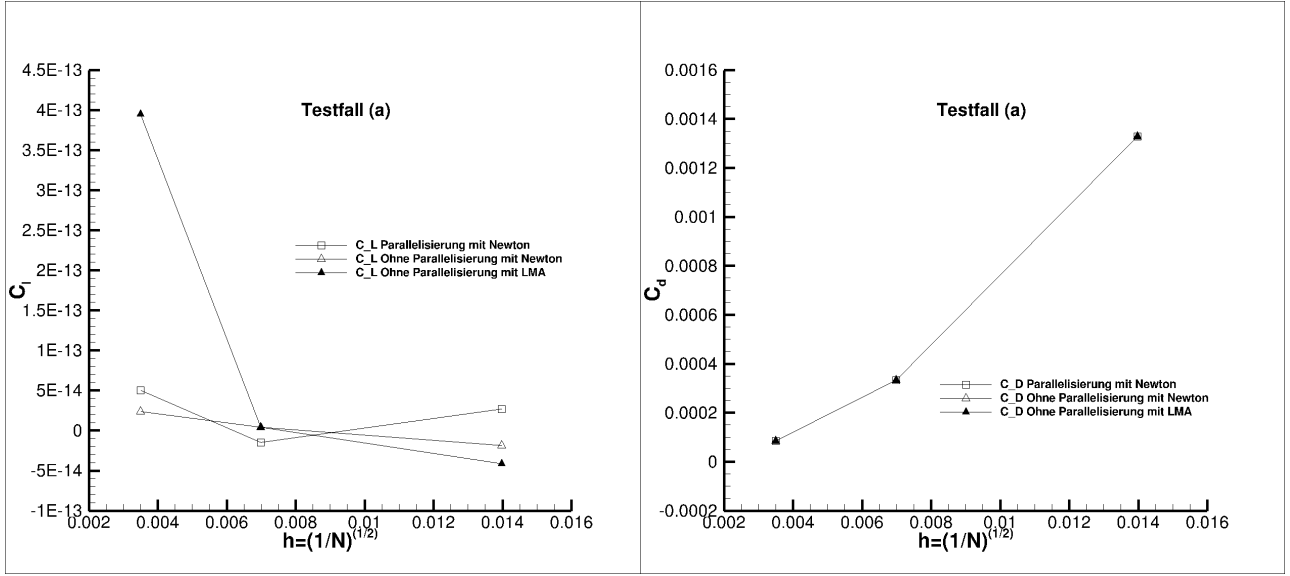


Abb. 4.7: Vergleich der Gitterkonvergenz des Auftriebsbeiwerts (Links) und Widerstandsbeiwerts (Rechts) in Vergleich zu Längeneinheit h , Eulere Strömung,

Die Abbildung (4.2) links zeigt den Konvergenzverlauf des Testfalls ohne Parallelisierung und rechts mit Parallelisierung. Bei Anwendung des Levenberg-Marquardt Algorithmus und dem Newton-Verfahren ist die Anzahl der Mehrgitterzyklen fast gleich bis zur Konvergenz, weil die Jacobian-Matrix $\frac{\partial \mathbf{R}}{\partial \mathbf{W}}$ quadratisch, full-Rank und positiv definiert sein sollte. Somit sind die beiden Lösungsmethode eine fast identische Mathematische Operation: Zur Berechnung der inversen Matrix der Linearesystem (3.46) bzw. (3.51). Die Berechnungszeit mit dem Levenberg-Marquardt Algorithmus ist jedoch deutlich länger als mit dem Newton-Verfahren, da die Berechnung mit dem Levenberg-Marquardt Algorithmus deutlich kompliziert ist als mit dem Newton-Verfahren, seit die Berechnung des Transposematritzen des Jakobians $(\frac{\partial \mathbf{R}}{\partial \mathbf{W}})^T$ durchführt. Allgemein sinkt die Konvergenzgeschwindigkeit mit steigender Gitterstufe auch mit beiden Testmethoden. Bei Parallelisierung werden nur die Testfälle mit dem Newton-Verfahren erfolgreich auskonvergiert. Die Lösung der Testfälle mit Levenberg-Marquardt Algorithmus mit Parallelisierung divergiert nach einer Zeitschritt der Berechnung. Im Vergleich zu den ohne Parallelisierung konvergieren die Lösungen nach dem Newton-Verfahren mit Parallelisierung ungefähr mit der Hälfte der Mehrgitter-Zyklus.

Im Folgenden geht das in Vergleich der Variable. Man kann in den Grafiken (4.4) - (4.6), die Konvergenzhistorie des Auftriebsbeiwerts und des Widerstandsbeiwerts bei drei Gittergrößen sehen. Der Auftriebsbeiwert bei allen ohne Parallelisierung Testfälle liegt ungefähr am 200 Mehrgitter-Zyklus und unterliegt relativ großen Schwankungen. Vor allem die Schwankungen der Konvergenzhistorie des Auftriebsbeiwerts ohne Parallelisierung durch Levenberg-Marquardt Algorithmus sind bei Feingittern sehr groß. Mit der steigenden Gitterstufe konvergiert der Widerstandsbeiwert aber in längeren Mehrgitter-Zyklen. Die Grafik (4.7) links zeigt die konvergierte Lösung des Auftriebsbeiwerts in allen Testfällen pro Einheitslänge des Gitters $h = N^{-\frac{1}{2}}$ und rechts des Widerstandsbeiwerts. Der Auftriebsbeiwert liegt in allen Testfällen sehr Nähe von $C_l = 0,0$ und die Differenzen des Werts sollen numerische Fehler sein, da die Werte $1e-13$ Magnituden sind. Außerdem ist es offensichtlich klar, dass der Auftriebsbeiwert $c_l = 0,0$ ist, da symmetrische Profile durch Nichtviskoseströmung mit $AoA = 0,0^\circ$ getestet wird. Die Konvergierte Lösung des Widerstandsbeiwerts pro h ist in der Grafik (4.7)

rechts zu sehen. Im Vergleich des Widerstandsbeiwerts kann man sehen, dass der Widerstandsbeiwert im Allgemeinen mit steigender Einheitslänge h immer ansteigt. Jedoch sind die konvergierten Widerstandsbeiwerte in allen Testfällen und in jeder Gitterstufe fast identisch. Daran ist zu erkennen, dass der Auftriebsbeiwert und der Widerstandsbeiwert in gleich großen durch je zwei Methoden konvergiert werden. Die Abbildung (4.8) - (4.11) zeigten die Druckverteilung auf dem NACA0012 Profil bei jeder Gitterstufe. Bei Grob- und Fein-Gitterstufen bestehen kleine Differenzen zwischen den ohne Parallelisierung mit dem Newton-Verfahren und des Levenberg-Marquardt Algorithmus bei $x/c = 0,08$.

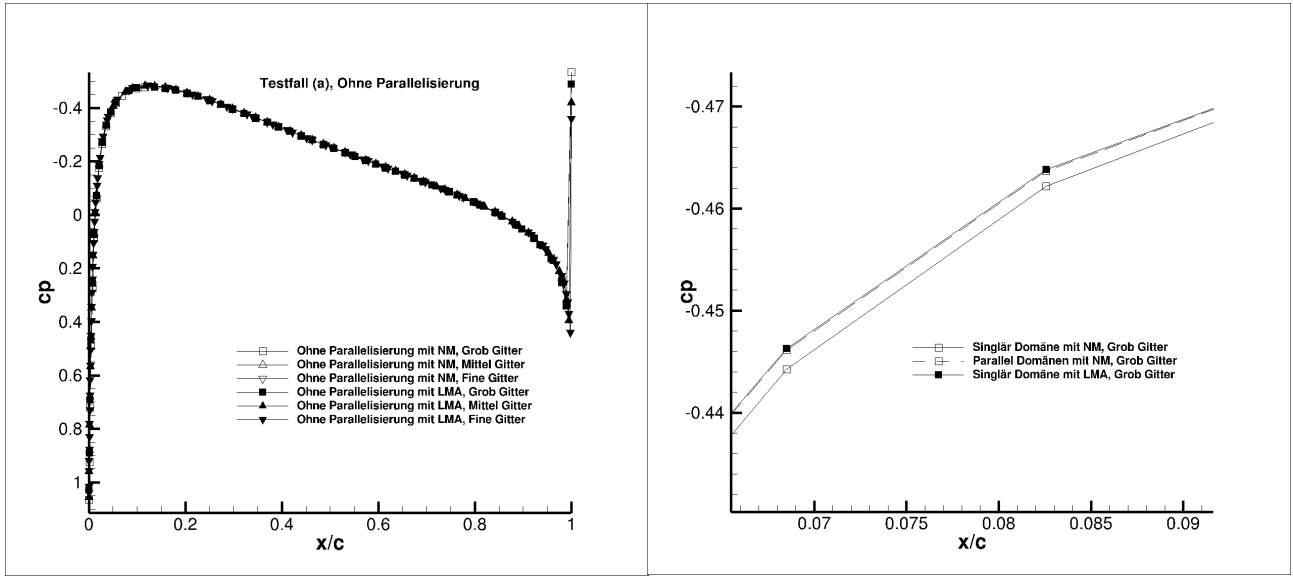


Abb. 4.8: Drückbeiwert c_p Verteilung auf dem NACA0012 Profile, eulere Strömung, ohne Parallelisierung, **Abb. 4.9:** Drückbeiwert c_p Verteilung auf dem NACA0012 Profile, eulere Strömung, grob Gitter,

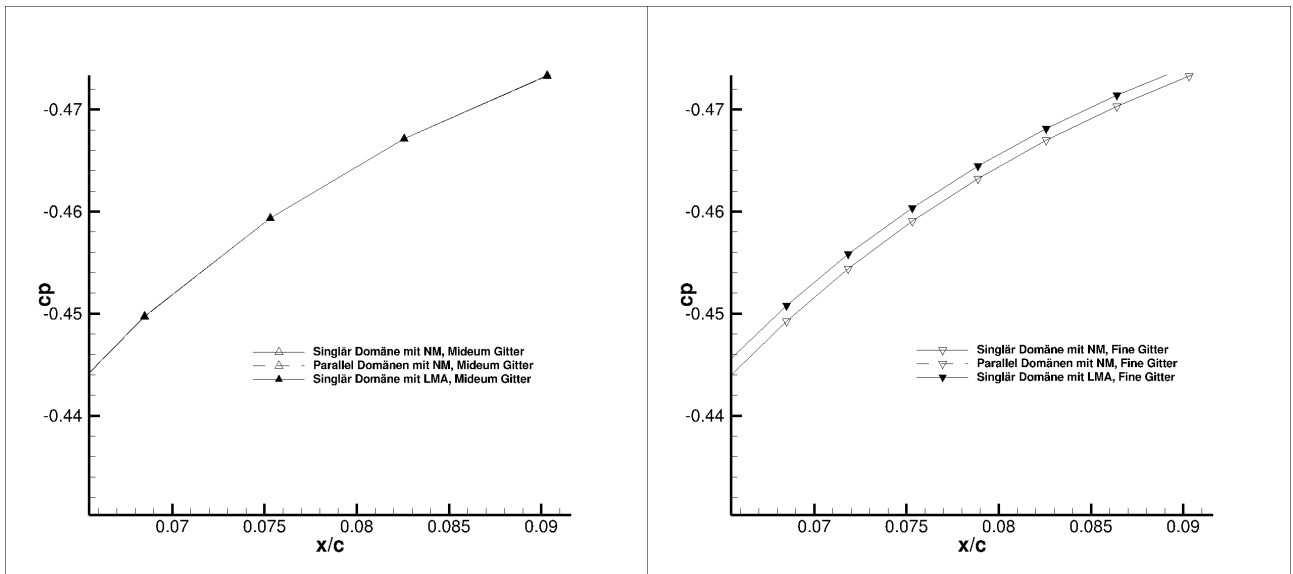


Abb. 4.10: Drückbeiwert c_p Verteilung auf dem NACA0012 Profile, eulere Strömung, mittel Gitter **Abb. 4.11:** Drückbeiwert c_p Verteilung auf dem NACA0012 Profile, eulere Strömung, fein Gitter

4.1.2 Testfall(b):NACA0012, Laminare Strömung

Wie im Testfall(a) werden in diesem Testfall die Lösung aller Testfällen durch das Newton-Verfahren ohne Parallelisierung und Parallelisierung konvergiert. Die Lösungen anhand des Levenberg-Marquardt Algorithmus könnten nicht in Grob Gitterfall mit Parallelisierung zur Konvergenz gehen.

Der algorithmischer Parameter findet sich in der Tabelle (4.5).

Parameter	Wert		
	Grob	Mittel	Fein
Gittergröße	128×64	256×128	512×256
Zahl der Zelle	24896	98944	394496
Zahl der Punkten	34052	133632	529408
Zahl der Domänen für Parallelisierung Test	3	4	6

Tab. 4.4: Gitter Data von NACA0012 auf dem Laminarströmung

Parameter	Wert		
	Grob	Mittel	Fein
Runge-Kutta Stufe	3	3	3
Initiale CFL-Zahl	10	10	10
Finale CFL-Zahl	1000	1000	1000
Start up steps with initial CFL number mean flow	10	10	10
Zahl der Gauss-Seidel Sweeps	5	5	5
Mehrgitterzyklus	4W	4W	4W
Relaxationsparameter	0,4	0,4	0,4

Tab. 4.5: Algorithmischer Parameter für NACA0012 Profile mit Laminar Strömung

Parameter	Wert		
	Grob	Mittel	fine
Gittergröß	128×64	256×128	512×256
Runge-Kutta Stufe	3	3	3
Initiale CFL-Zahl	10	10	10
Finale CFL-Zahl	1000	1000	1000
Start up steps with initial CFL number mean flow	5	5	5
Relaxationsparameter	0,6	0,6	0,6
Zahl der Gauss-Seidel Sweeps	5	25	25
Mehrgitterzyklus	4W	4W	4W

Tab. 4.6: Algorithmischer Parameter für NACA0012 Profile mit Laminar Strömung mit Levenberg-Marquardt Algorithmus

Die Grafik (4.12) stellt die Konvergenzhistorie der ohne Parallelisierung links und der Parallelisierung rechts im Vergleich zu allen konvergierten Fällen durch das Newton-Verfahren und durch den Levenberg-Marquardt Algorithmus dar. Die Lösung durch beiden Methoden ohne Parallelisierung konvergiert mit der Zahl der ähnlichen Mehrgitter-Zyklus, nämlich bei Grob Gitter 800 und bei Mittel Gitter 2200. Bei Parallelisierung sieht anders aus. Die Lösungen nach Newton-Verfahren in allen Gitterstufen werden unter 200 Mehrgitter-Zyklus konvergiert, während Konvergenz der Lösung durch Levenberg-Marquardt Algorithmus bei Mittel Gitter 1700 und bei Fein Gitter 2000 Mehrgitter Zyklus. In Testfall(b) kann man bestätigen, dass Konvergenz der Lösung durch Levenberg-Marquardt Algorithmus sehr langsam ist, weil der Korrekturwert für jeden Iterationsschritt klein ist im Fall großer Regularisierungsparameter γ_n (diskutiert in Kapitel 3.3). Die Langsamkeit des konvergenz durch Levenberg-Marquardt Algorithmus kann in dem Grafik(4.14), das Vergleich das Konver-

genzhistorie des Auftriebsbeiwerts und Widerstandsbeiwerts durch Newton-Verfahren und Levenberg-Marquardt Algorithmus ohne Parallelisierung und mit Parallelisierung bei Mittel und Fein Gitter ist, gesehen werden. Die beiden Fällen bis ca. 400 Mehrgitter-Zyklus schwakt sich sehr groß als nach Newton-Verfahren. Konvergenzhistorie des Auftriebsbeiwert und des Widerstandsbeiwerts pro ein Einheitslänge h in Grafiken (4.17) sieht man wesentliche Unterschiede zwischen ohne Parallelisierung und Parallelisierungsfälle in Testfall (b). Jedoch hat fast keine Unterschiede zwischen nach Newton-Verfahren und Levenberg-Marquardt Algorithmus. In dem Bild (4.18) sieht man die Vergleich der Drückverteilung auf dem NACA0012 Profile. Da gibt es keine Unterschiede der Ergebnis mit den zwei Lineresystemn. Vergleich des Reibungskoefficients von den Ergebnissen durch Newton-Verfahren und Levenberg-Marquardt Algorithmus ist in der Bild (4.22). In der Nähe von Position auf dem Profile $x/c = 0,7$ sieht man ein Punktsprung, bei grob-, mittel- und feiner Gitter am $x/c = 0,62$, $x/c = 0,72$, $x/c = 0,76$ sind, weil diese Punkte die Strömung von der Profile ablöst. Die Ablösungspunkten hat maximal circa 15 % Bereich der Flugfläche auf jeden Gittersufe und je feinere Gitter ist desto verschiebt die Ablösungspunkt zu hinterkante Richtung. Die Ergebnis durch Levenberg-Marquardt Algorithmus wird auch gleiche Punkt gezeigt wie die durch Newton-Verfahren. Das heißt, dass die Lösungen mit jeden zwei Lineresystem in eine gleiche Lösung kovergiert werden. Am $x/c = 0,01$ der Reibungsbeiertverteilung in der Bild (4.22) liegen maximale c_f in jeden Gitterstufe. Während des Erreichung $c_f = 0,146$ am $x/c = 0,007$ in Feinste Gitter ist das in Mitteligitter $c_f = 0,144$ am $x/c = 0,007$ und in Grobgitter $c_f = 0,133$ am $x/c = 0,011$. Circa 8% Unterschiede zwischen drei Verschiedene Gitterstufe. Die Losung durch Levenberg-Marquardt Algorithmus zeigt Stömungslinie an der Hinterkante von NACA0012 in Bild (4.25). Das Zirkulationsgebiete an der Hiterkante wird gezeigt.

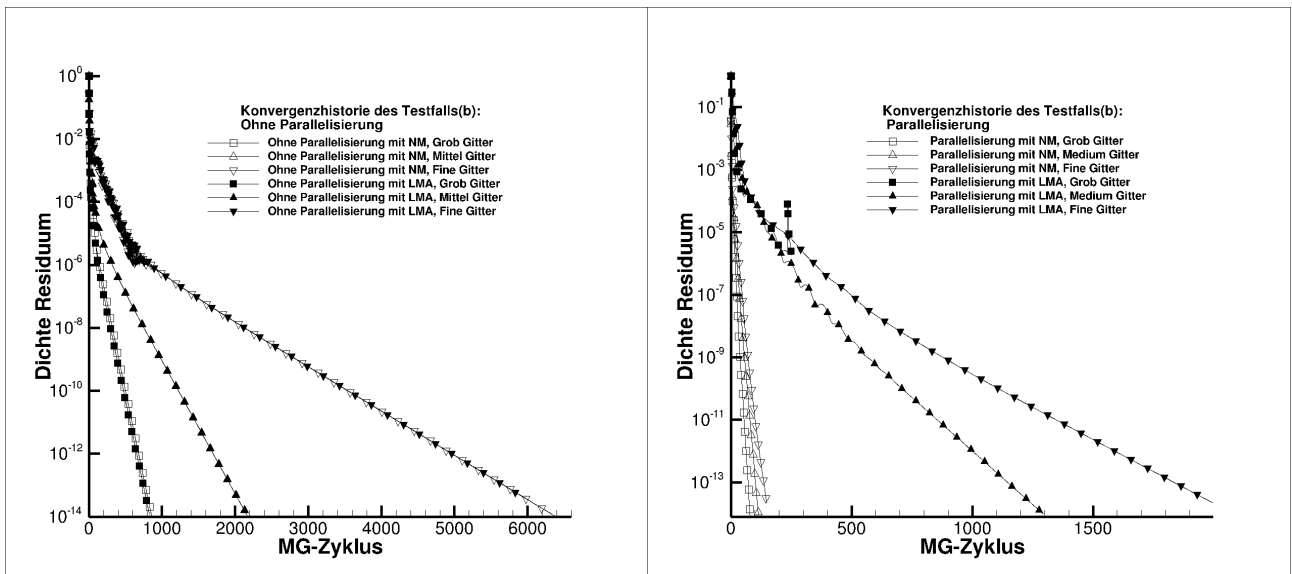


Abb. 4.12: Konvergenzhistorie des ohne Parallelisierung Testfällen (Links) und mit Parallelisierung Testfälle (Rechts), Laminare Strömung

Im Allgemein wird das bestätigt, dass die Berechnungen wesentlich länger pro Iteration dauern, wenn man Levenberg-Maquart Algorithmus in der Lösungsmethode implemientiert, als Newton-Verfahren, da zur Berechnen der trasponierte Jacobimatrix $\left(\frac{d\mathbf{R}}{d\mathbf{W}}\right)^T$ mehr Zeit und Speicher verbraucht.

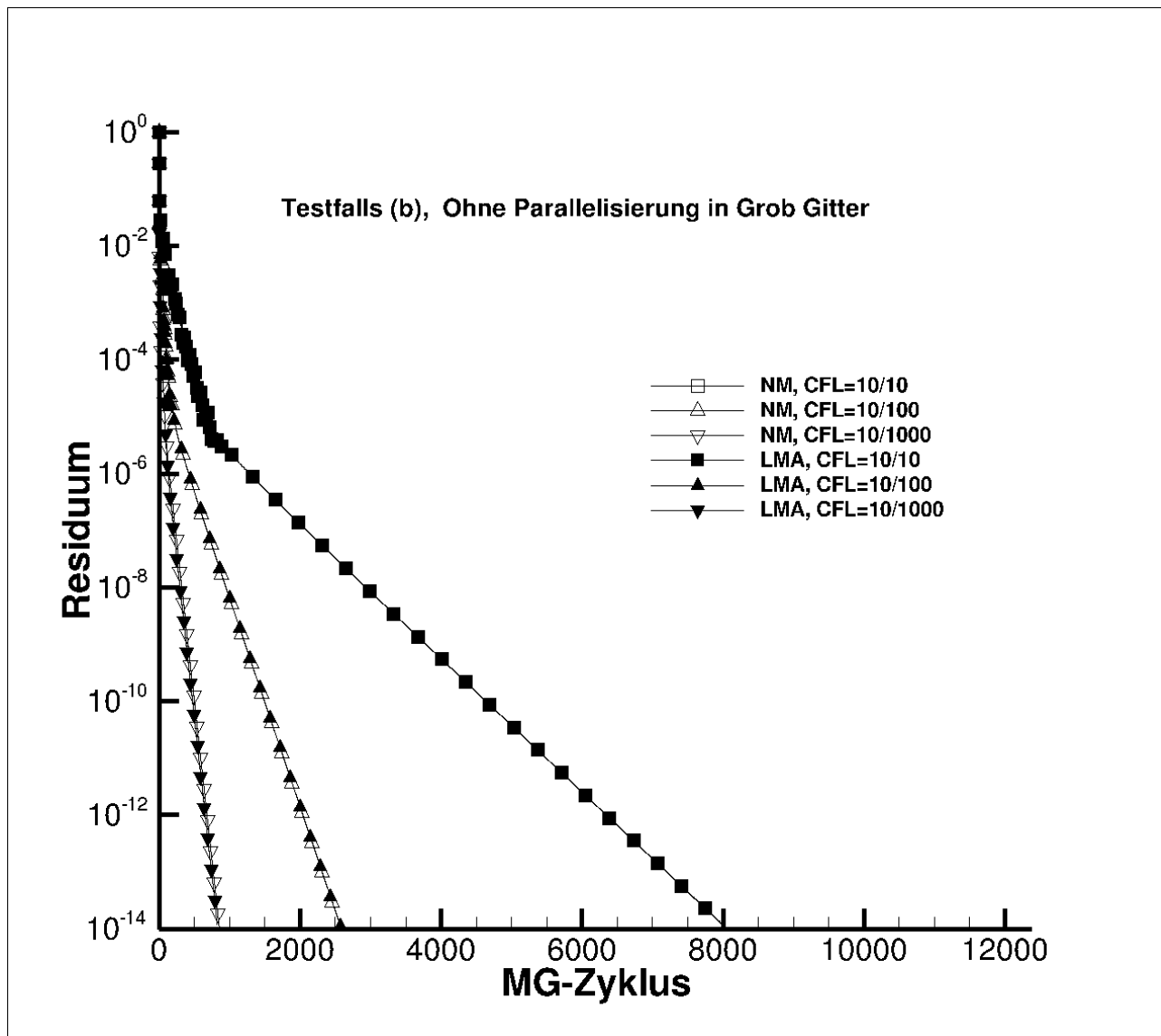


Abb. 4.13: Konvergenzhistorie ohne Parallelisierung Testfälle in Vergleich zu verschiedene CFL-Zahlen auf dem grob Gitter, Laminare Strömung

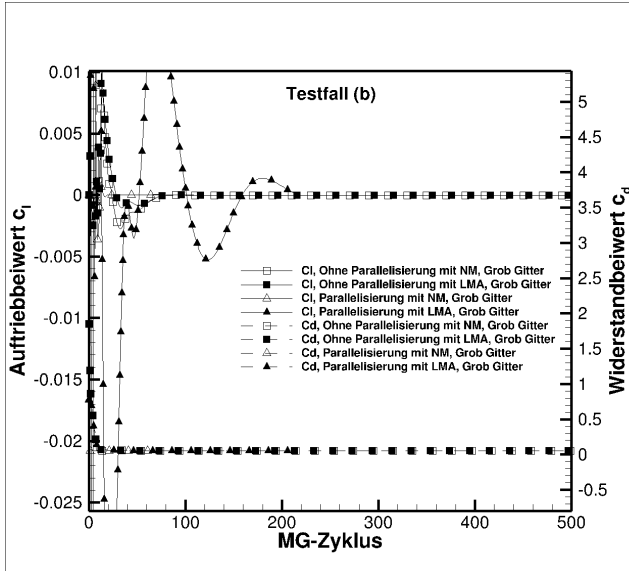


Abb. 4.14: Vergleich der Konvergenzhistorie von Widerstandsbeiwert, Laminare Strömung, Grob Gitter

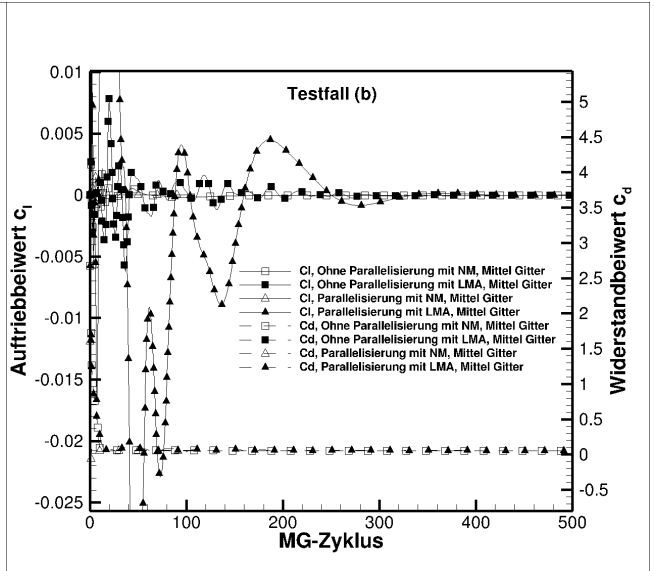


Abb. 4.15: Vergleich der Konvergenzhistorie von Auftriebsbeiwert, Laminare Strömung, Mittel Gitter

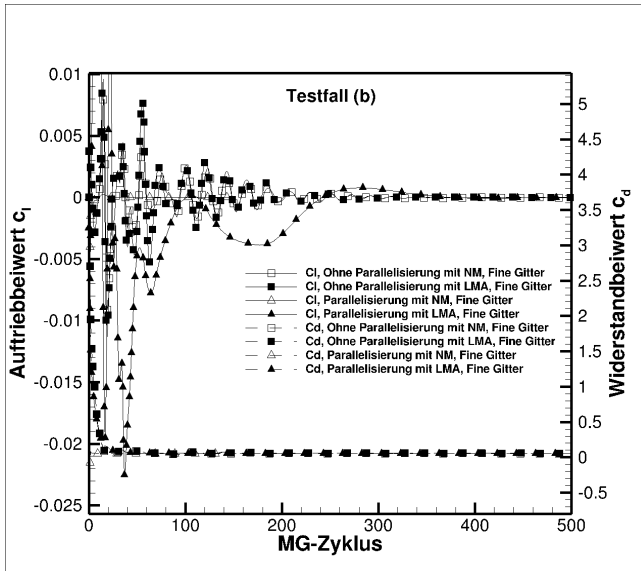


Abb. 4.16: Vergleich der Konvergenzhistorie von Auftriebsbeiwert, Laminare Strömung, fine Gitter

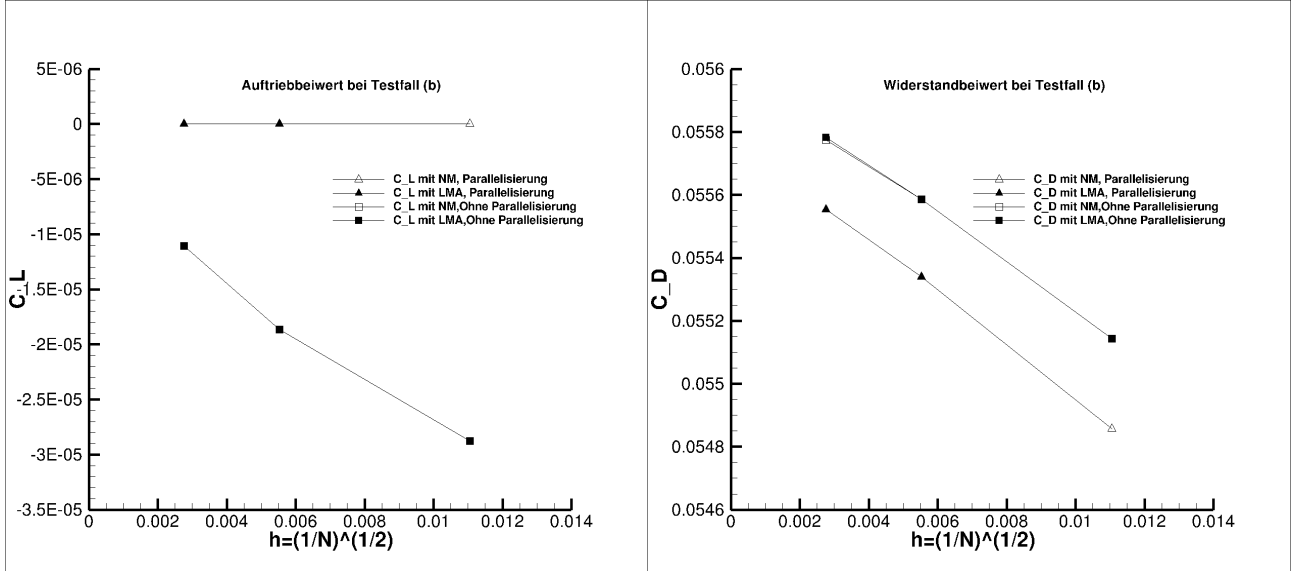


Abb. 4.17: Vergleich der Gitterkonvergenz des Auftriebsbeiwerts c_l (Links) und des Widerstandsbeiwerts c_d (Rechts) in Vergleich zu Längeneinheit h , laminare Strömung

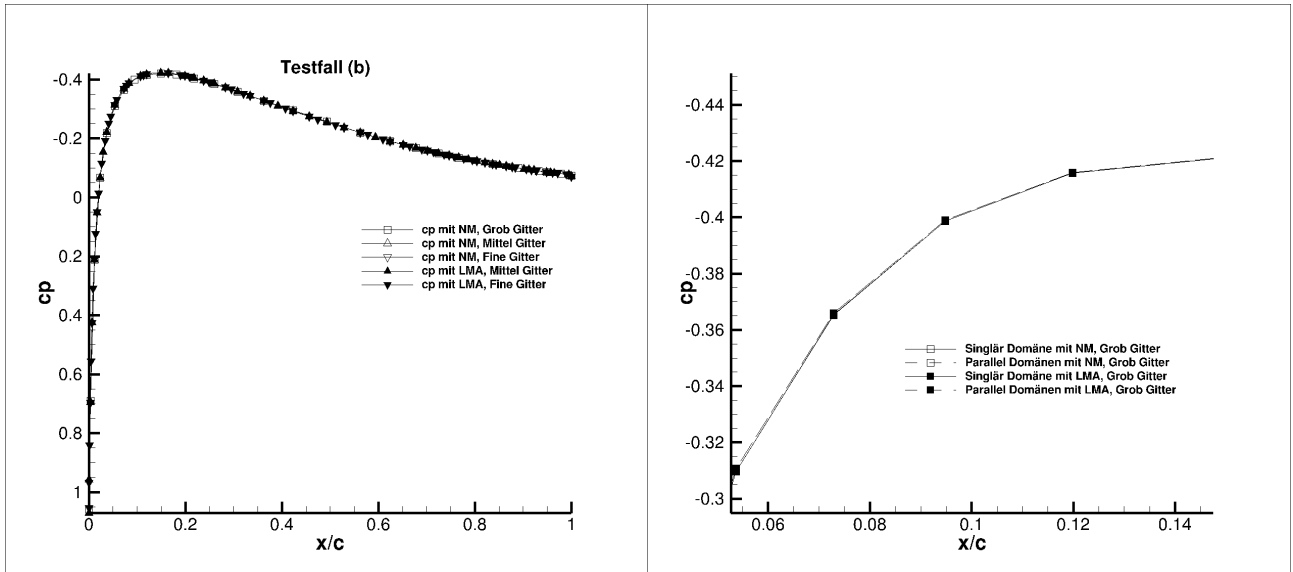


Abb. 4.18: Drückbeiwert c_p Verteilung ohne Parallelisierung auf dem NACA0012 Profile in Vergleich zu Gitterstufe zwischen NM und LMA, laminare Strömung,

Abb. 4.19: Drückbeiwert c_p Verteilung mit Parallelisierung und ohne Parallelisierung auf dem NACA0012 Profile in Vergleich zu NM und LMA, laminare Strömung, grob Gitter,

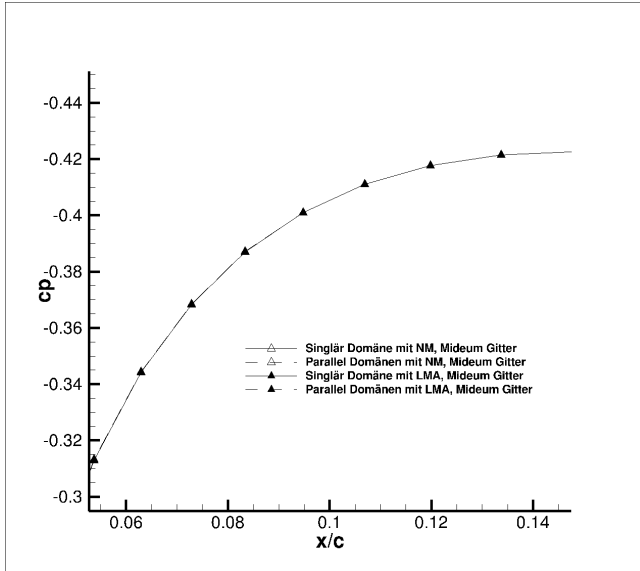


Abb. 4.20: Drückbeiwert c_p Verteilung mit Parallelisierung und ohne Parallelisierung auf dem NACA0012 Profile in Vergleich zu NM und LMA, laminare Strömung, mittel Gitter,

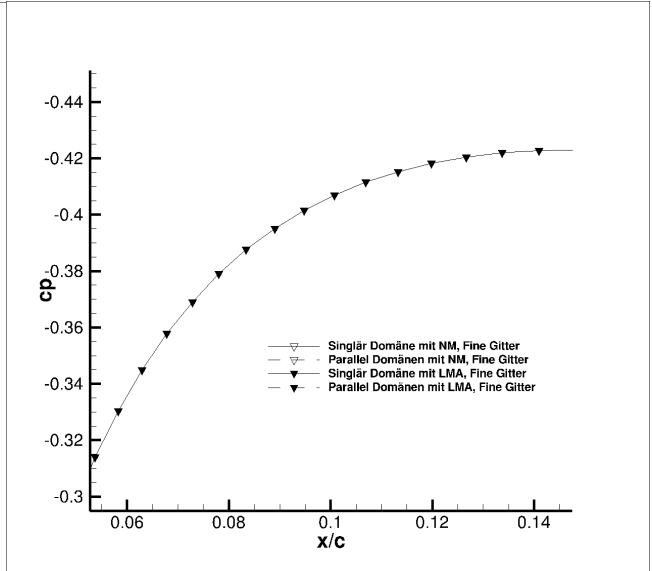


Abb. 4.21: Drückbeiwert c_p Verteilung mit Parallelisierung und ohne Parallelisierung auf dem NACA0012 Profile in Vergleich zu NM und LMA, laminare Strömung, fein Gitter

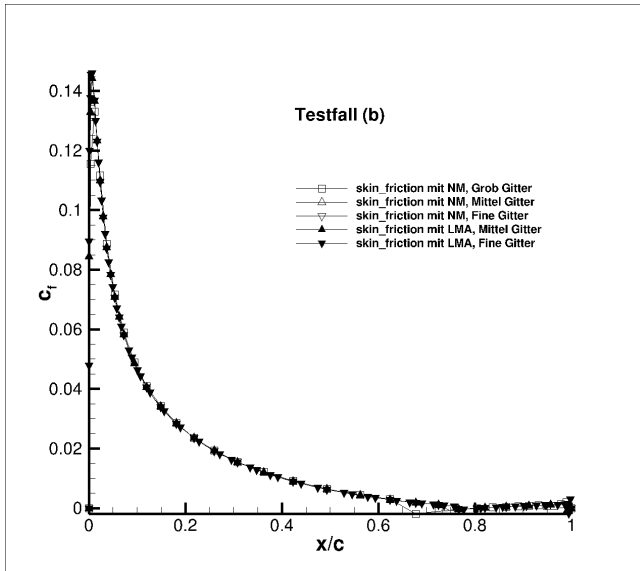


Abb. 4.22: Reibungsbeiwert c_f Verteilungen auf dem NACA0012 in Testfall(b), in Vergleich zu Gitterstufe zwischen NM und LMA, $\text{AoA} = 0^\circ$

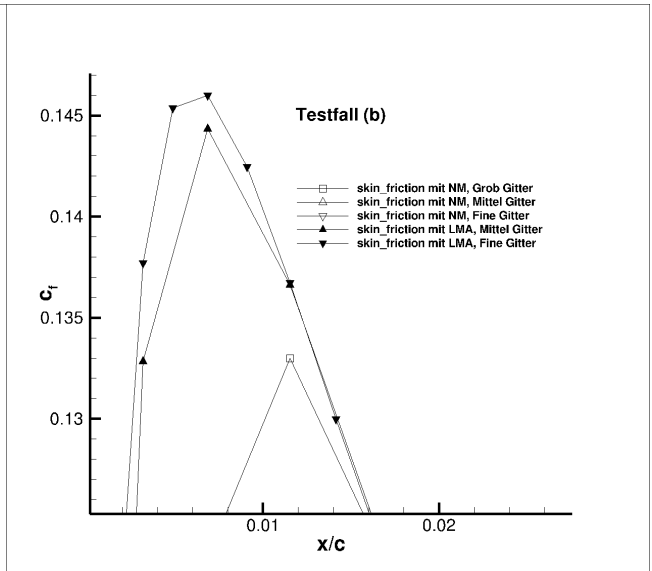


Abb. 4.23: Reibungsbeiwert c_f Verteilungen auf dem NACA0012 in Testfall(b), in Vergleich zu Gitterstufe zwischen NM und LMA, $\text{AoA} = 0^\circ$

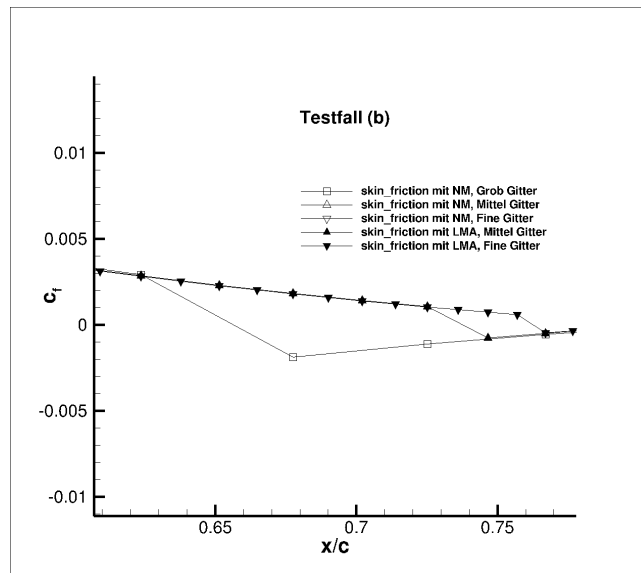


Abb. 4.24: Reibungsbeiwert c_f Verteilungen auf dem NACA0012 in Testfall(b), in Vergleich zu Gitterstufe zwischen NM und LMA, $AoA = 0^\circ$

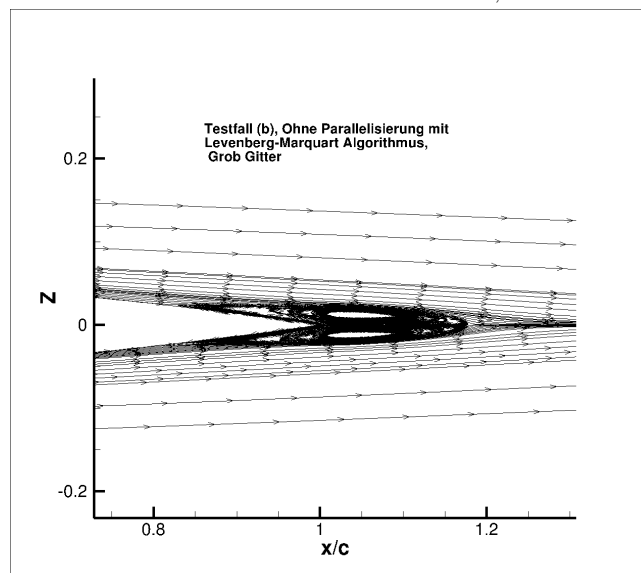


Abb. 4.25: Stömungslinie an der Hinterkante von NACA0012

4.2 3D-Testfälle

4.2.1 Testfall(c):Hemisphere Zylinder

Für den dreidimensionalen Fall wird ein Zylinder mit hemisphärischen Enden. Mit dem Levenberg-Marquardt Algorithmus konnten für die dreidimensionalen Testfälle keine Konvergenzen erzielt werden. Jedoch werden die Ergebnisse mit dem Newton-Verfahren für Glätter der implizit Runge-Kutta Verfahren gezeigt. Die Geometrie des Zylinders wurde wie im Experiment [20] gewählt. Am Fernfeld wird ein Rieman invarianttische Randbedingung gestellt. Außerdem darf am externen Rand freie Ausströmung sein, die auf input Machzahl basiert. An der Zylinderwand wird ein adiabatischer Zustand angenommen. Im Experiment beträgt der Durchmesser des Zylinders 1 Zoll und die Länge 10 Zoll. Somit wird referenz Bereich 10 genommen. Im Abbildung (4.26) Rechts wird das zylindrische Gittermodell und Links Fernfeld des Gitters dargestellt. In dieser Abschlussarbeit werden drei verschiedene Gitterstufen getestet. Die getesteten Gitter haben eine Block- und tetraedrische Struktur. In Tabelle (4.7) sind die verwendeten Gittergrößen aufgelistet. Zum Vergleich der Testergebnisse werden für alle Gitterstufen die gleiche CFL-Zahl ausgewählt. Im Allgemeinen wird in der Startphase der Iteration eine kleine CFL-Zahl gewählt. Für diesen Versuch wird die CFL-Zahl nach dem Kriterium (4.2) ausgewählt. Für den Testfall (c) wird der CFL-Startwert auf 2 und der Endwert auf 50 gesetzt. Im Fall einer zu groß gewählten CFL-Zahl konnte keine Konvergenz erzielt werden. Die Parameter für den Zylinderversuch sind in Tabelle 4.8 aufgelistet. Mehrgitter Zyklus wird 2V für die beide von mittlere- und turbulenzen Gleichung genommen, da die Lösung bei finere Gitter mit Mehrgitter Zyklus 3V oder 4W nicht konvergiert werden. Höhere Mehrgitter Zyklus beschleunigt das Konvergenz während langsame Konvergenz in Generel mit 2V. Als Turbulenzmodell wird der negative Spalart-Allmaras ein-Gleichungsmodell verwendet, der unter Abschnitt (2.1.2) vorgestellt wurde. In diesem Versuch werden mit fünf verschiedene AoA =n gerechnet $0,0^\circ$; $5,0^\circ$; $10,0^\circ$; $15,0^\circ$, und $19,0^\circ$. Als Referenz werden die physikalischen Werte der freien Strömung verwendet: $T_{ref} = 300K$, Reynolds-Zahl $Re = 0,35e6$ und Machzahl $Ma = 0,35$. Testfall (c) ist Parallelberechnung mit der Anzahl der Domänen für Grob-, Mittel- und Feingitter 24, 48 und 192.

Parameter	Wert		
Gitter Stufe	Grob	Mittel	Fein
Zahl der Tetrahedra	723456	5732352	45637632
Zahl der Punkten	128851	988067	7736131

Tab. 4.7: Gitter Data von Hemisphere-Zylinder

Parameter	Wert		
Gitterstufe	Grob	Mittel	Fein
Runge-Kutta Stufe	3	3	3
Initiale CFL-Zahl	2	2	2
Finale CFL-Zahl	50	50	50
Zahl der Gauss-Seidel Sweeps	5	5	5
Mehrgitterzyklus	2V	2V	2V
Mehrgitterzyklus für Turbulenz Gleichung	2V	2V	2V
Relaxationsparameter	1,0	1,0	1,0
Anstieg Faktor für CFL Nummer turbulenz Strömung	1.1	1.1	1.1
Linear solver turbulenz Strömung Gleichungen	7	7	7
Entropy fix Delta für Lösungsalgorithm Turblenz Strömung	0.1	0.1	0.1
Zahl der Turblenzschritt pro mittleren Durchflussschritt	5	5	5
Zahl der Domäne	24	48	192

Tab. 4.8: Algorithmischer Parameter für Hemispher-Zylinder

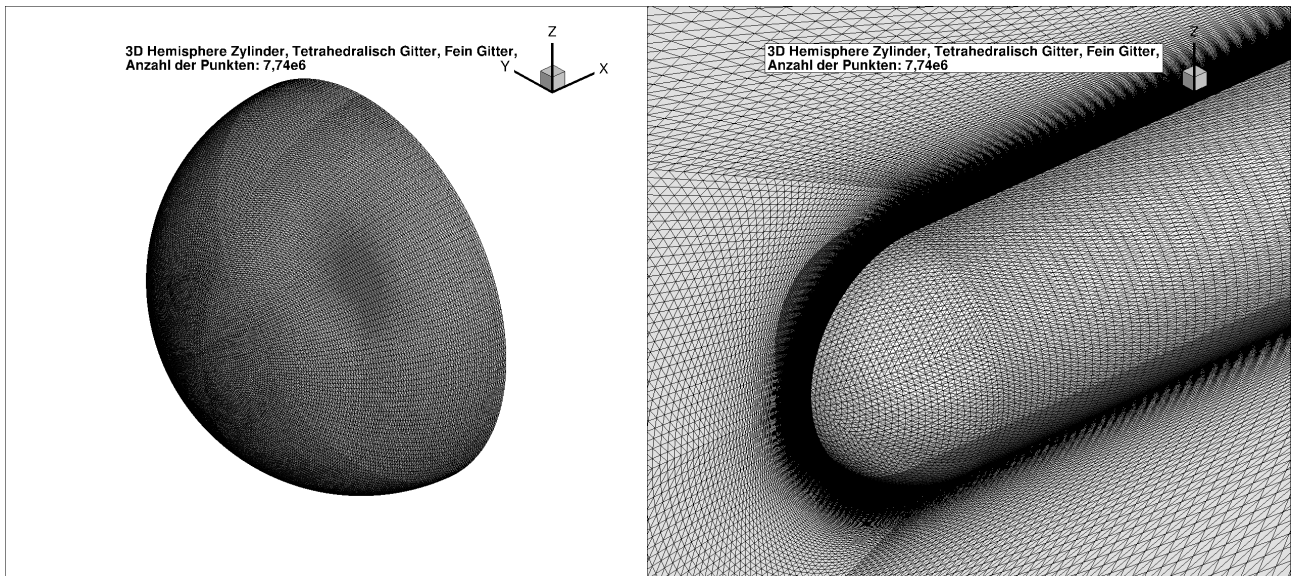


Abb. 4.26: Fernfeld des Gitters für Hemisphere Zylinder (Links) und in der Nähe von Hemisphere Zylinder (Rechts), fein Gitterstufe

Der Konvergenzverlauf bei einem $\text{AoA} = \text{von } 0,0^\circ$ ist in Abbildung (4.27) dargestellt. Die Linie stellt Konvergenzhistorie von mittlere Strömung und die unterbrochene Linie stellt Konvergenzhistorie von turbulenz Strömung dar. Die Konvergenzhistorie von allen Gitterstufen sehen ähnliche aus und sind fast keine Unterschiede zwischen allen $\text{AoA} = 0 - 19^\circ$. In Abbildung (4.28) werden die Verläufe der Druckbeiwerte über die Zylinderoberfläche bei einem $\text{AoA} = \text{von } 0,0^\circ$ dargestellt. Größere Unterschiede zeigen sich bei $x/c = 0,4$ (Abbildung (4.29)) und $x/c = 10$ (Abbildung (4.30)). Die ausgefüllten Punkte sind Messwerte aus dem Experiment [20]. Einerseits trifft der Verlauf des Druckbeiwerts mit feinem Gitter die Punkte aus dem Experiment relativ gut, aber es gibt einen sehr großen Unterschied zwischen den minimalen Druckbeiwerten für das feine und grobe Gitter. Der Unterschied beträgt etwa 30%. Darüber hinaus liegen die minimalen Druckbeiwerte für das feine und mittel Git-

ter sehr nahe beieinander, aber das Minimum für das grobe Gitter liegt etwa 20% stromabwärts. An der Hinterkante nehmen alle Druckbeiwerte ab. Am Hinterkante wurden die Druckbeiwerte bei allen Gittern gesunken. Besonders wurde der Druckbeiwert bei Feingitter von $c_p = -0,01$ bei $x = 8,5$ bis $c_p = -0,05$ bei $x = 10$, als andere Gittern, da Randbedingung bei Feingitter starker beeinflusst hat, als andere Gitter, und diese Abweichung am Hinterkante physikalisch keine Bedeutung hat. In Abbildung (4.31) wird der Verlauf des Reibungsbeiwerts über die Zylinderoberfläche in x-Richtung dargestellt. Die Reibungsbeiwerte zwischen Ober- und Unterseite unterscheiden sich fast über die gesamte Zylinderlänge um $c_{f,x} = 0,02$. Die maximalen Reibungsbeiwerte liegen alle bei etwa $x = 0,3$, allerdings liegen zwischen den Maximalwerten bis zu 30% unterschied.

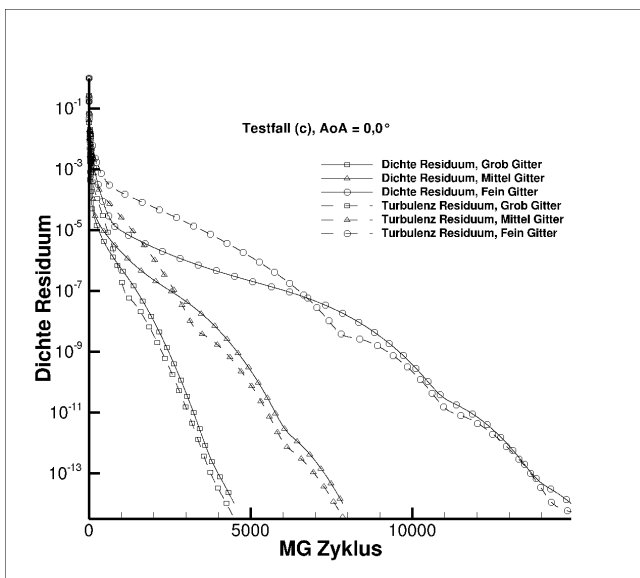


Abb. 4.27: Konvergenzhistorie in Vergleich zu Gitterstufe, $AoA = 0^\circ$

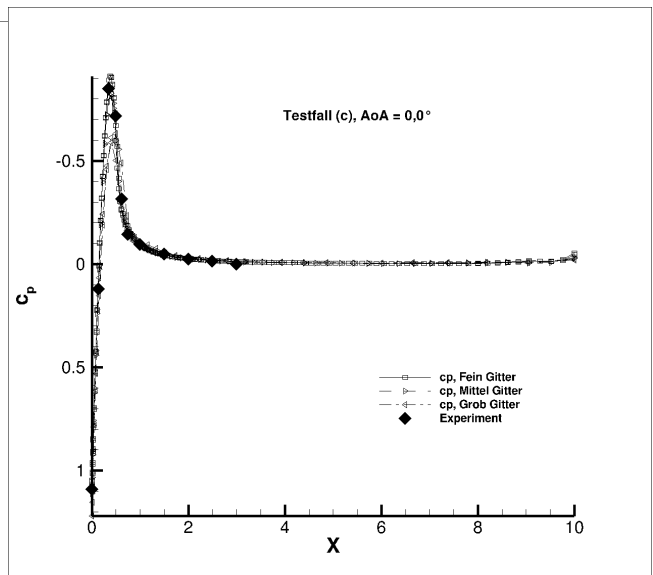


Abb. 4.28: Druckbeiwert c_p Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, $AoA = 19^\circ, AoA = 0^\circ$

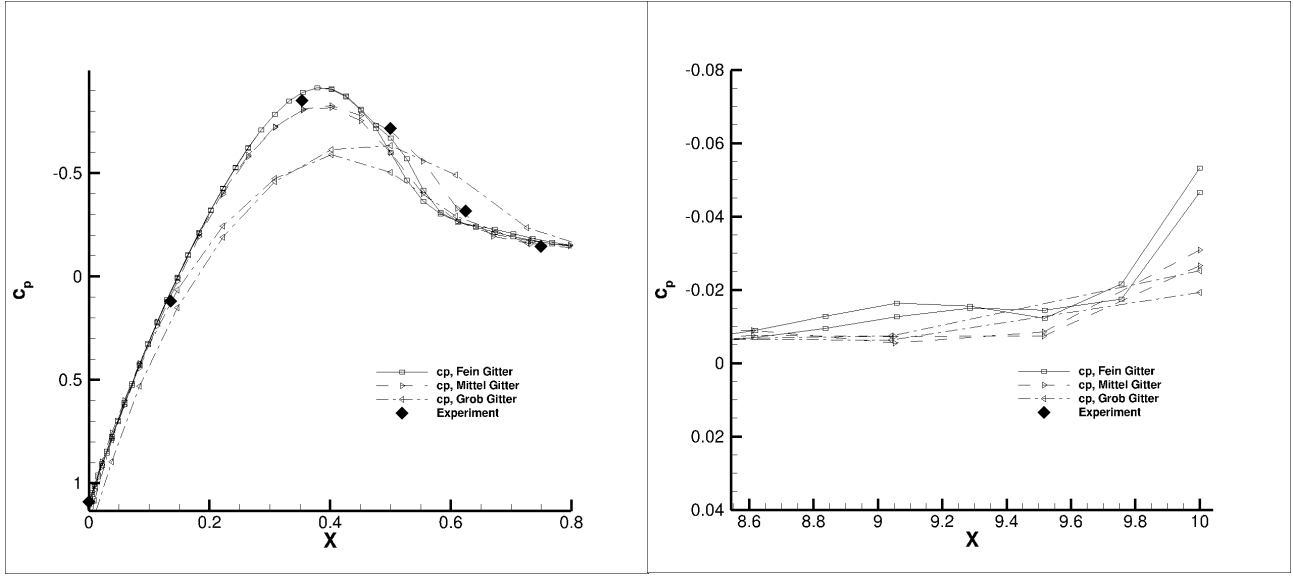


Abb. 4.29: Drückbeiwert c_p Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, AoA = 0°

Abb. 4.30: Drückbeiwert c_p Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, AoA = 0°

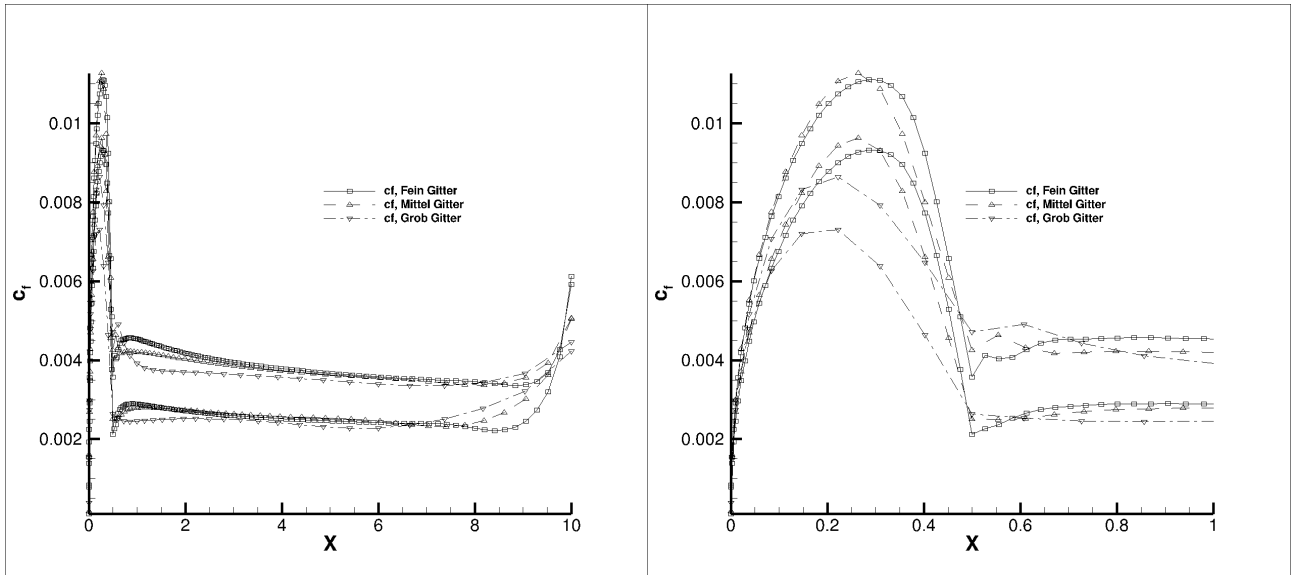


Abb. 4.31: Reibungsbeiwert c_f Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, AoA = 0°

Abb. 4.32: Reibungsbeiwert c_f Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, AoA = 0°

In den Abbildungen (4.33) ist der konvergierten Auftriebsbeiwert c_l und Widerstandsbeiwert c_d für den AoA = $0, 0^\circ$ abgebildet. Der Abszissenwert h beschreibt eine Längeneinheit für dreidimensionale Gitter $h = \left(\frac{1}{N}\right)^{\frac{1}{3}}$. Der Widerstandsbeiwert c_d wird auch Reibungswiderstand $c_{d,v}$ und Drückwiderstand $c_{d,p}$ separat dargestellt, die in Abbildung (4.34) sich ein Zusammenhang zwischen der Gittergröße und der Annäherung an den Grenzwert erkennen lässt. Je feiner das Gitter ist, desto näher konvergiert

der Auftriebsbeiwert c_l gegen null. Reibungswiderstandsbeiwert $c_{d,v}$ steigt zunehmende Längeneinheit h . In Gegenteil sinkt Drückwiderstandsbeiwert $c_{d,p}$ mit steigenden Längeneinheit h . Insgesamt ist die Abweichung der Werte bei fein und mittel Gitterstufe kleiner als mit groben Gitterstufe. Bei allen getestete AoA =n wird die Werte $c_d, c_{d,v}$ und $c_{d,p}$ ähnliche Verhältnis bestätigt. Auftriebsbeiwert c_l hat spezifische Verhältnis mit steigenden AoA = verfolgt. Vor allem ist ab AoA = $5,0^\circ$ der Wert bei mittel Gitter in generell kleiner als bei andere Gitter in Abbildung (A.1), (A.3), (A.5) und (A.7) . Bei AoA = $15, 19^\circ$ ist Abweichung der Wertunterschiede zwischen bei fein und grob Gitter kleiner als zwischen fein- und Mittel, da das Mitteligitter vermutlich komplexe Strömungsverhältnis wegen solchen hohen AoA =n nicht präzise Auftriebsbeiwert berechnen kann.

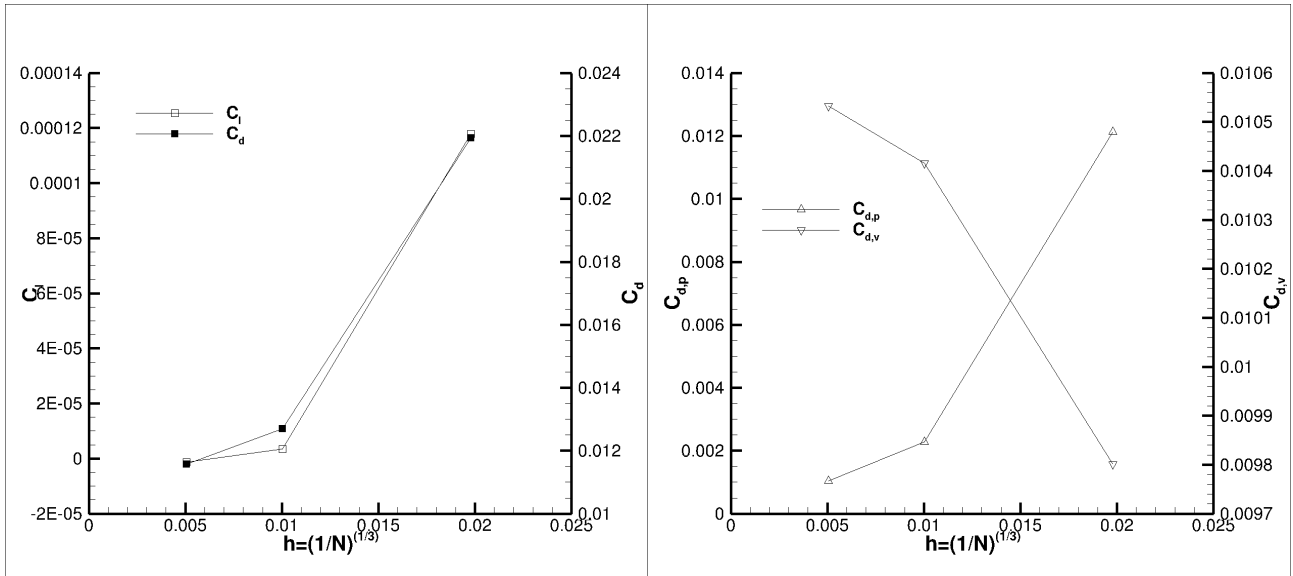


Abb. 4.33: Auftriebsbeiwert c_l und Widerstandsbeiwert c_d Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Längeneinheit h , AoA = 0°

Abb. 4.34: Drückwiderstandsbeiwert $c_{d,p}$ und Reibungswiderstandsbeiwert $c_{d,v}$ Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Längeneinheit h , AoA = 0°

5 Zusammenfassung

Zum approximativen Lösen der kompressiblen Reynolds gemittelten Navier-Stokes Gleichungen (RANS) wird ein agglomerätes FAS-Mehrgitter-Schema verwendet. Als Glätter des Mehrgitterverfahren kommt das mehrstufiges implizites Runge-Kutta-Verfahren zur Anwendung. Da ein lineares Gleichungssystem in jeder Runge-Kutta-Stufe gelöst werden muss, wird die Gauss-Seidel Verfahren zur Lösen der lineares Gleichungssystem verwendet.

Wie bereits in der Einleitung und der Übersicht erwähnt, kann die Lösung durch die Newtonische Methode nur in der Nähe der exakten Lösung und nur lokal konvergiert werden. Darüber hinaus wird das Newton-Verfahren nur verwendet, wenn bekannt ist, dass die Initialschätzung im Testfall in der Nähe der exakte Lösung ist. Ziel dieser Arbeit war es, den schlechten Zustand der Newtonischen Methode zu eliminieren. Eine Idee hierfür war die Optimierung des Newton-Verfahrens mithilfe des Levenberg-Marquardt Algorithmus. In dieser Arbeit wurde statt eines Newtonische Verfahrens der Levenberg-Marquardt Algorithmus implementiert und getestet. Die Ergebnisse wurden in den 2D-Testfälle mit dem Newton-Verfahren verglichen. Daraus ergaben sich folgende Erkenntnisse:

- In den 2D Testfall (a) wird die Lösung mit Parallelisierung nicht konvergiert. Jedoch werden die Lösungen ohne Parallelisierung erfolgreich konvergiert.
- In den 2D Testfällen (a)(b) werden die Lösungen durch die Lösungsmethode mit dem Levenberg-Marquardt Algorithmus mit $CFL_{final} = 10$ bis $1,000$ konvergiert.
- In dem 2D-Testfall (b) mit Parallelisierung werden die Lösungen durch die Lösungsmethode mit dem Levenberg-Marquardt Algorithmus wesentlich längere Mehrgitter Zyklus genommen.
- Im 2D-Testfällen (a)(b) erhält man ein selbe Ergebnis des Auftriebsbeiwerts und Widerstandsbeiwerts mit dem Levenberg-Marquardt Algorithmus und mit dem Newton-Verfahren.
- Im 2D-Testfällen (a)(b) hat keine Vorteil Levenberg-Marquardt Algorithmus zu verwenden.
- Die Berechnung dauert wesentlich länger pro Iteration, wenn man Levenberg-Maquart Algorithmus in der Lösungsmethode imprementiert, als Newton-Verfahren.
- Im 3D-Testfall(c) wurde der Hemisphere Zylinder mit turbulenzen Fluss durch Lösungsmethoden mit dem Newton-Verfahren erfolgreich getestet, die auf negativ Spalart-Almars Turbulenzenmodell basiert.

Prinzipiell soll die Lösungsmethode mit dem Levenberg-Marquardt Algorithmus robuster sein als Newton-Verfahren. Allerdings konnten es im Rahmen dieser Arbeit zu wenige Testfälle durchgeführt werden, um die Robustheit der Lösungsmethode mit dem Levenberg-Marquardt Algorithmus wirklich zu bestätigen. Dafür müssen zum Beispiel Turbulenzströmungsfälle und\oder noch feinere Gitterstufen getestet werden. Zur Durchführung der Tests wird die Parallelisierung der Lösungsmethode benötigen.

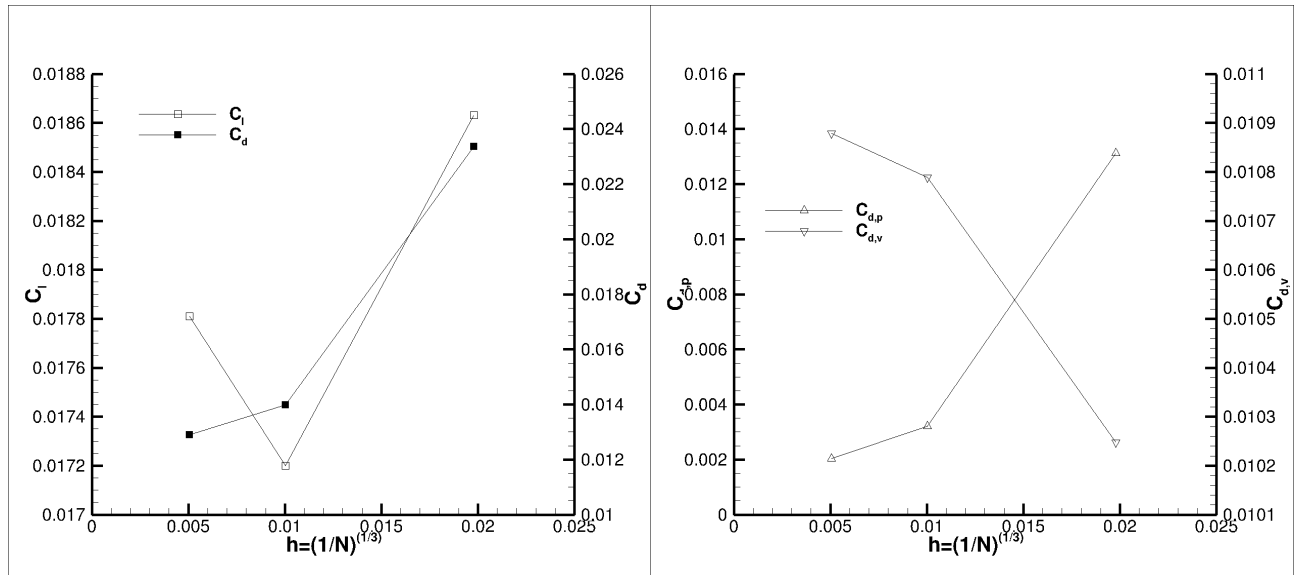
Literaturverzeichnis

- [1] Antony Jameson, Wolfgang Schmidt, and Eli Turkel. Numerical solution of the euler equations by finite volume methods using runge kutta time stepping schemes. In *14th fluid and plasma dynamics conference*, page 1259, 1981.
- [2] Seokkwan Yoon and Antony Jameson. Lower-upper symmetric-gauss-seidel method for the euler and navier-stokes equations. *AIAA journal*, 26(9):1025–1026, 1988.
- [3] Niles A Pierce and Michael B Giles. Preconditioned multigrid methods for compressible flow calculations on stretched meshes. *Journal of Computational Physics*, 136(2):425–445, 1997.
- [4] Cord-Christian Rossow. Efficient computation of compressible and incompressible flows. *Journal of Computational Physics*, 220(2):879–899, 2007.
- [5] RC Swanson, Eli Turkel, and C-C Rossow. Convergence acceleration of runge–kutta schemes for solving the navier–stokes equations. *Journal of Computational Physics*, 224(1):365–388, 2007.
- [6] RC Swanson and C-C Rossow. An efficient solver for the rans equations and a one-equation turbulence model. *Computers & Fluids*, 42(1):13–25, 2011.
- [7] Cord C Rossow. Convergence acceleration for solving the compressible navier-stokes equations. *AIAA journal*, 44(2):345–352, 2006.
- [8] Stefan Langer. Investigation and application of point implicit runge–kutta methods to inviscid flow problems. *International Journal for Numerical Methods in Fluids*, 69(2):332–352, 2012.
- [9] Stefan Langer. Application of a line implicit method to fully coupled system of equations for turbulent flow problems. *International Journal of Computational Fluid Dynamics*, 27(3):131–150, 2013.
- [10] Stefan Langer. Hierarchy of preconditioning techniques for the solution of the navier-stokes equations discretized by 2nd order unstructured finite volume methods. In *ECCOMAS 2012 Conference, Conference Proceeding Series*, 2012.
- [11] Stefan Langer, Axel Schwöppe, and Norbert Kroll. Investigation and comparison of implicit smoothers applied in agglomeration multigrid. *AIAA Journal*, 53(8):2080–2096, 2015.
- [12] PRaA Spalart and S1 Allmaras. A one-equation turbulence model for aerodynamic flows. In *30th aerospace sciences meeting and exhibit*, page 439, 1992.
- [13] Steven R Allmaras and Forrester T Johnson. Modifications and clarifications for the implementation of the spalart-allmaras turbulence model. In *Seventh international conference on computational fluid dynamics (ICCFD7)*, pages 1–11, 2012.
- [14] Peter Eliasson, Sofia Eriksson, and Jan Nordström. Influence of weak and strong solid wall boundary conditions on the convergence to steady-state of the navier-stokes equations. In *19th AIAA Computational Fluid Dynamics*, page 3551. 2009.
- [15] Stefan Langer. Agglomeration multigrid methods with implicit runge–kutta smoothers applied

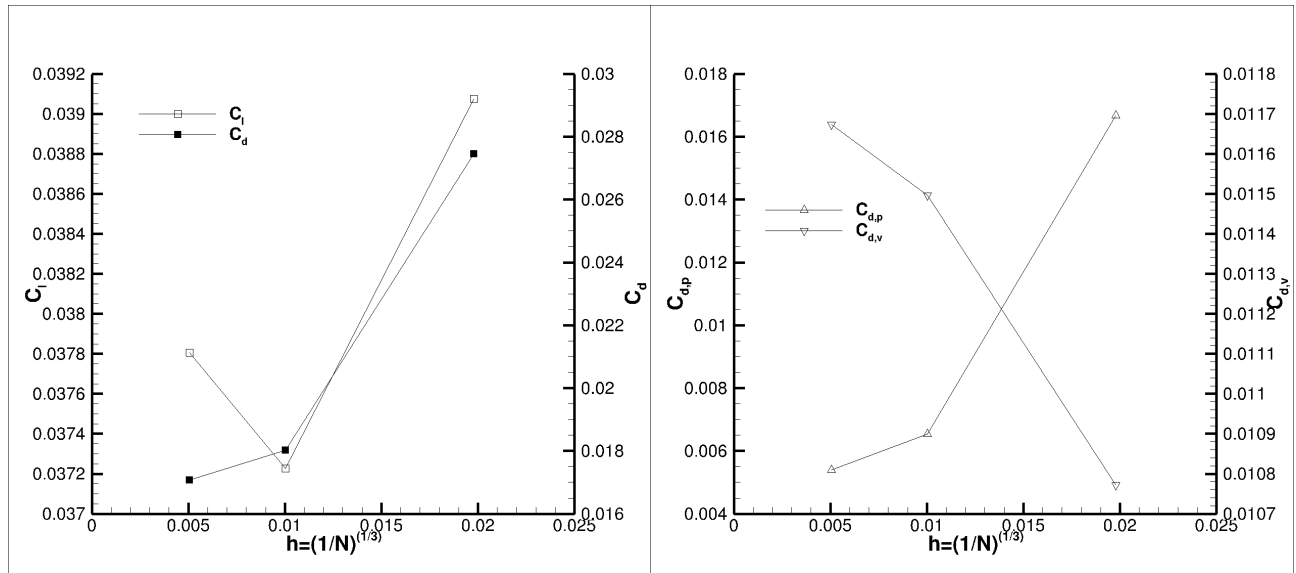
- to aerodynamic simulations on unstructured grids. *Journal of Computational Physics*, 277:72–100, 2014.
- [16] Ulrich Trottenberg and Anton Schuller. *Multigrid*. Academic Press, Inc., Orlando, FL, USA, 2001.
 - [17] Dimitri J Mavriplis. Multigrid techniques for unstructured meshes. Technical report, INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING HAMPTON VA, 1995.
 - [18] Yousef Saad. *Iterative methods for sparse linear systems*, volume 82. siam, 2003.
 - [19] K Madsen, HB Nielsen, and O Tingleff. Methods for non-linear least squares problems., 2004.
 - [20] T Hsieh. An investigation of separated flow about a hemisphere-cylinder at 0-to 19-deg incidence in the mach number range from 0.6 to 1.5. Technical report, ARNOLD ENGINEERING DEVELOPMENT CENTER ARNOLD AFB TN, 1976.
 - [21] RC Swanson and Stefan Langer. Steady-state laminar flow solutions for naca 0012 airfoil. *Computers & Fluids*, 126:102–128, 2016.

Anhang

Aerodynamische Kräfte bei Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$, $\text{AoA} = 5, 0^\circ$



Aerodynamische Kräfte bei Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$, $\text{AoA} = 10, 0^\circ$



Aerodynamische Kräfte bei Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$, $\text{AoA} = 15,0^\circ$

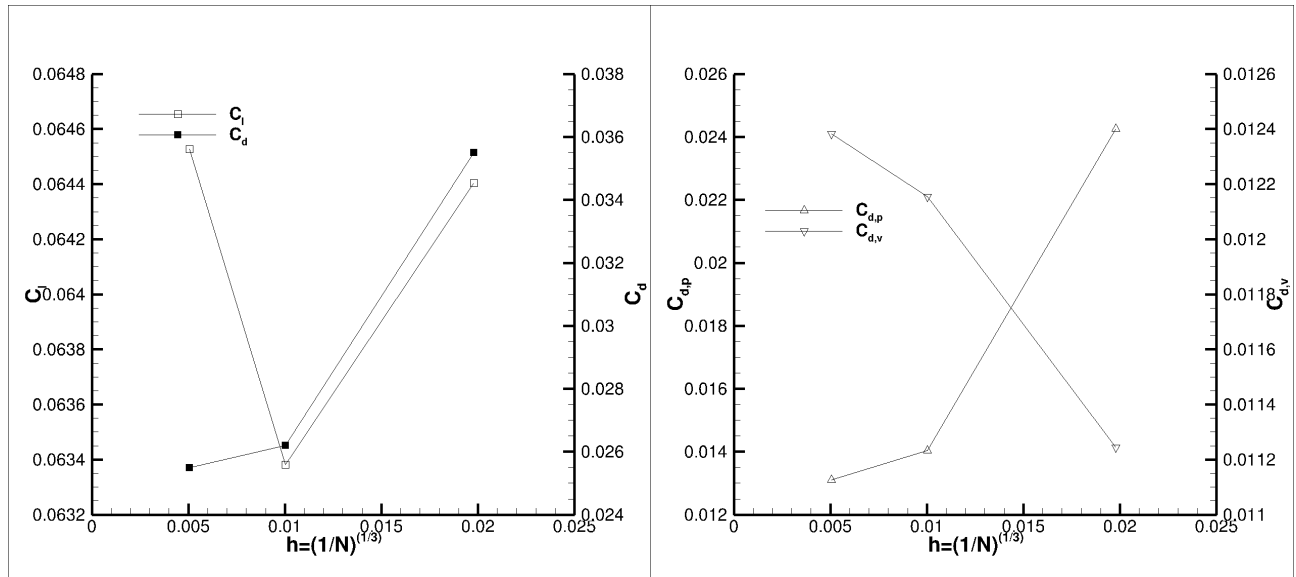


Abb. A.5: Auftriebsbeiwert c_l und Widerstandsbeiwert c_d Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Längeneinheit h , $\text{AoA} = 15^\circ$

Abb. A.6: Drückwiderstandsbeiwert $c_{d,p}$ und Reibungswiderstandsbeiwert $c_{d,v}$ Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Längeneinheit h , $\text{AoA} = 15^\circ$

Aerodynamische Kräfte bei Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$, $\text{AoA} = 19,0^\circ$

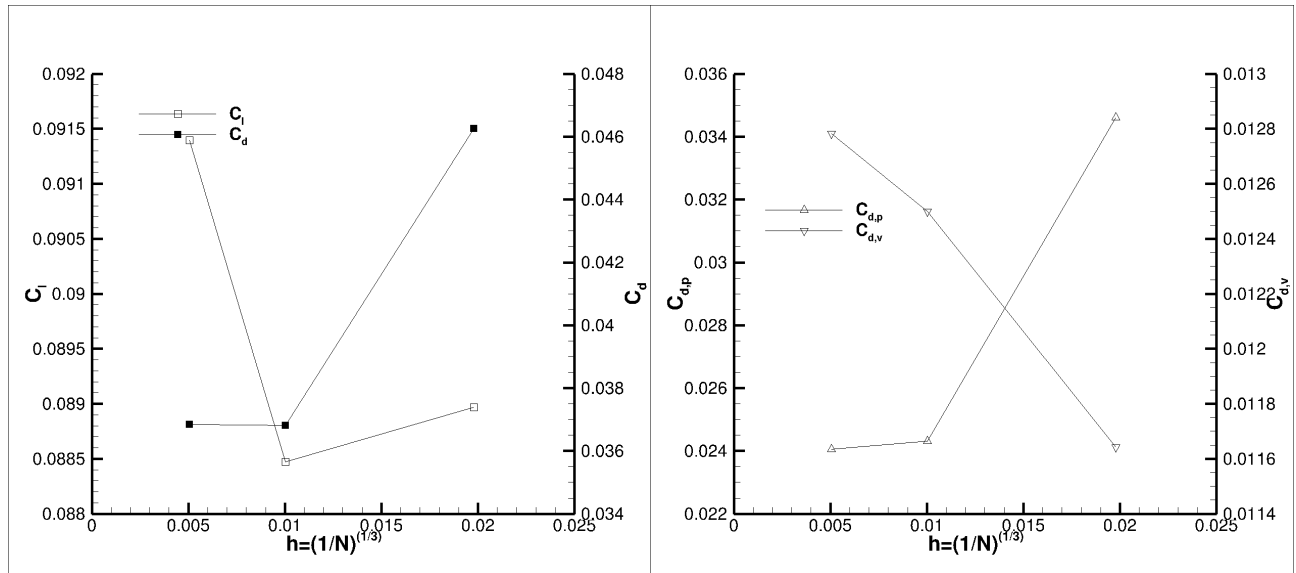


Abb. A.7: Auftriebsbeiwert c_l und Widerstandsbeiwert c_d Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Längeneinheit h , $\text{AoA} = 19^\circ$

Abb. A.8: Drückwiderstandsbeiwert $c_{d,p}$ und Reibungswiderstandsbeiwert $c_{d,v}$ Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Längeneinheit h , $\text{AoA} = 19^\circ$

Konvergenzhistorie und konvergierte Beiwerten bei Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$, AoA = $5,0^\circ$

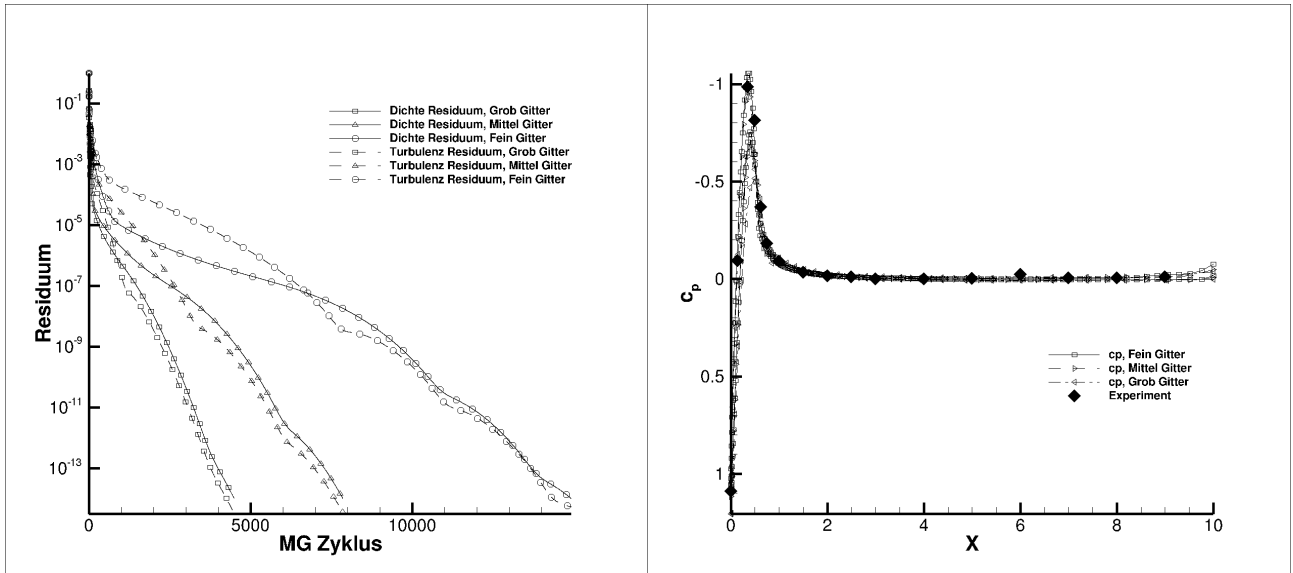


Abb. A.9: Konvergenzhistorie von Dichte und Turbulenz in Vergleich zu Gitterstufe, AoA = 5°

Abb. A.10: Druckbeiwert c_p Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, AoA = 5°

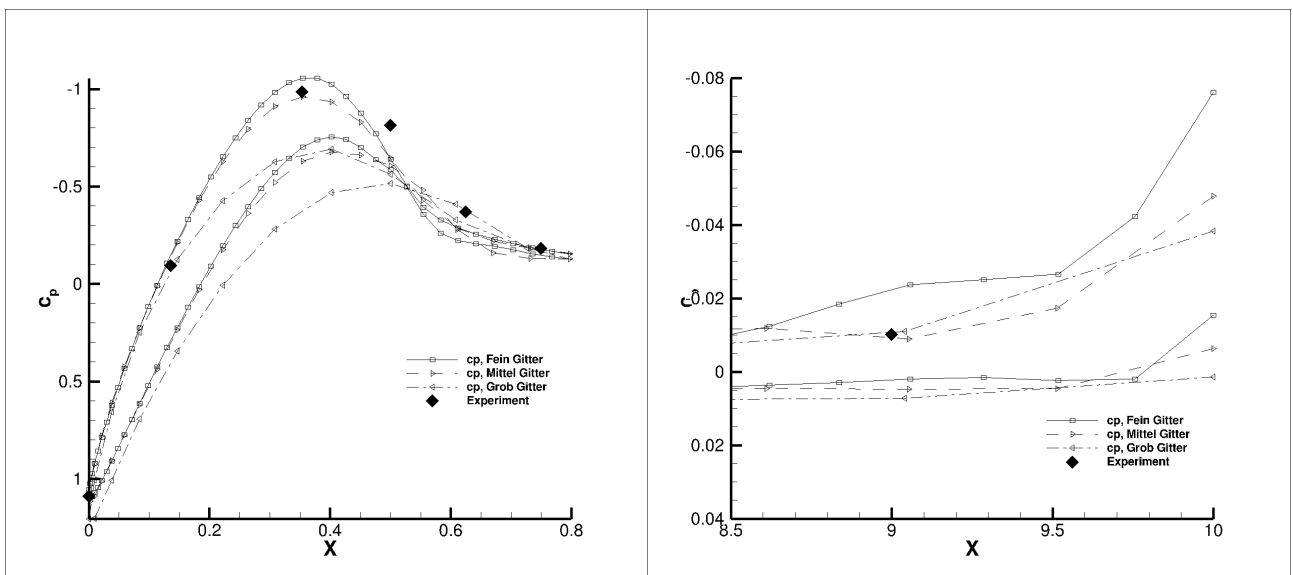


Abb. A.11: Druckbeiwert c_p Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, AoA = 5°

Abb. A.12: Druckbeiwert c_p Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, AoA = 5°

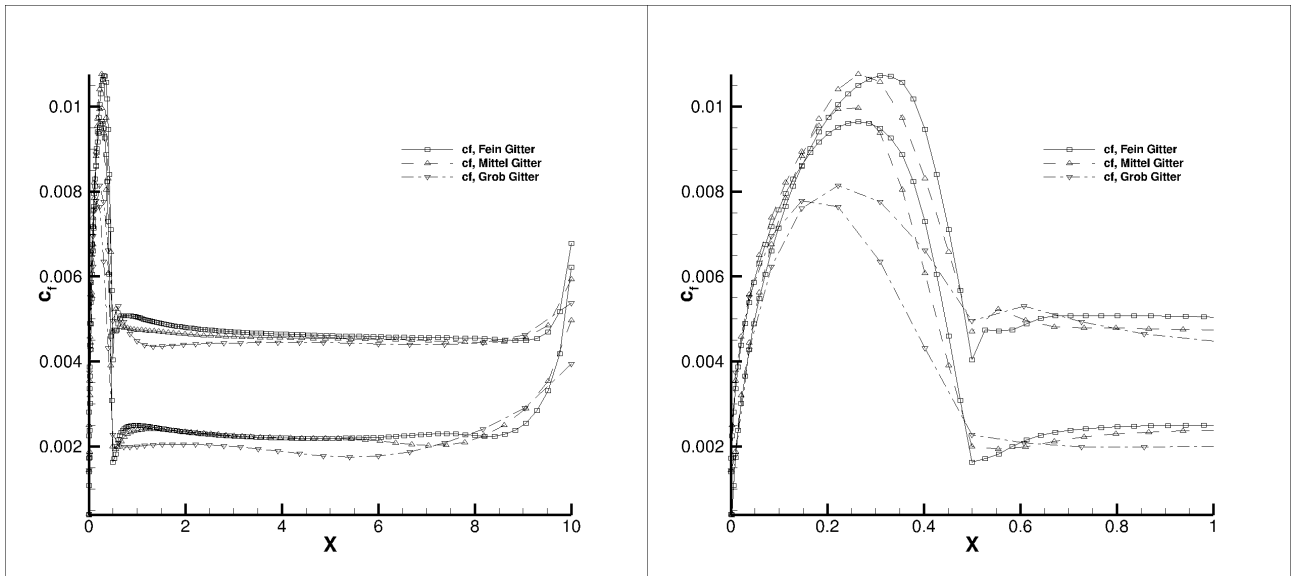


Abb. A.13: Reibungsbeiwert c_f Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, AoA = 5°

Abb. A.14: Reibungsbeiwert c_f Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, AoA = 5°

Konvergenzhistorie und konvergierte Beiwerten bei Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$, $AoA = 10, 0^\circ$

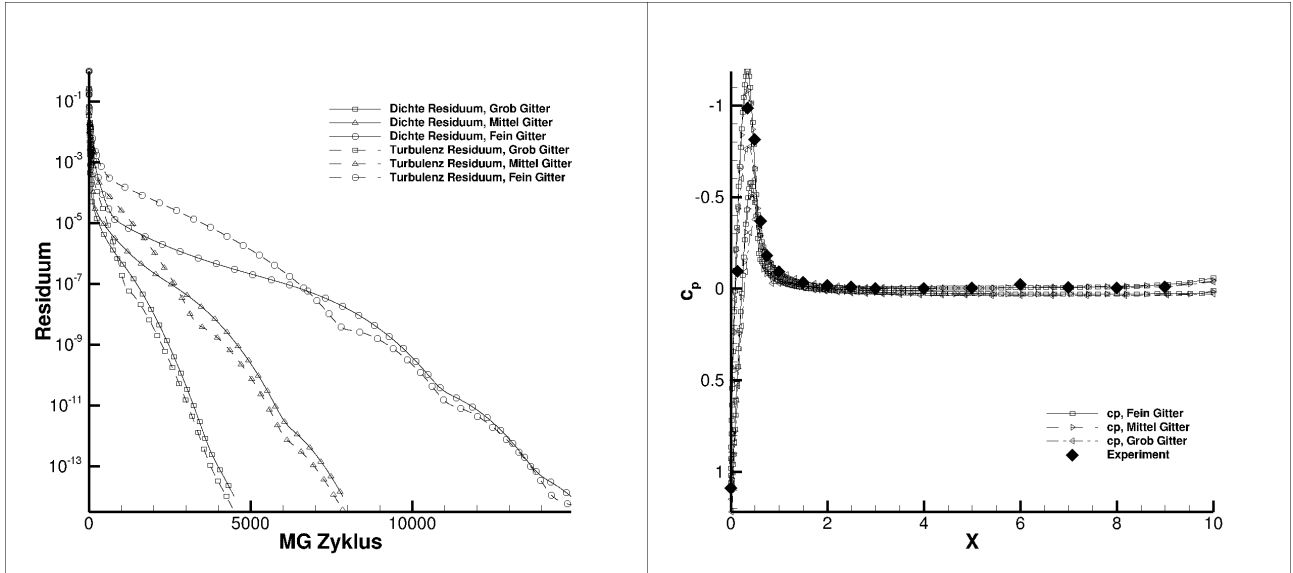


Abb. A.15: Konvergenzhistorie von Dichte und Turbulenz in Vergleich zu Gitterstufe, $AoA = 10^\circ$

Abb. A.16: Drückbeiwert c_p Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, $AoA = 10^\circ$

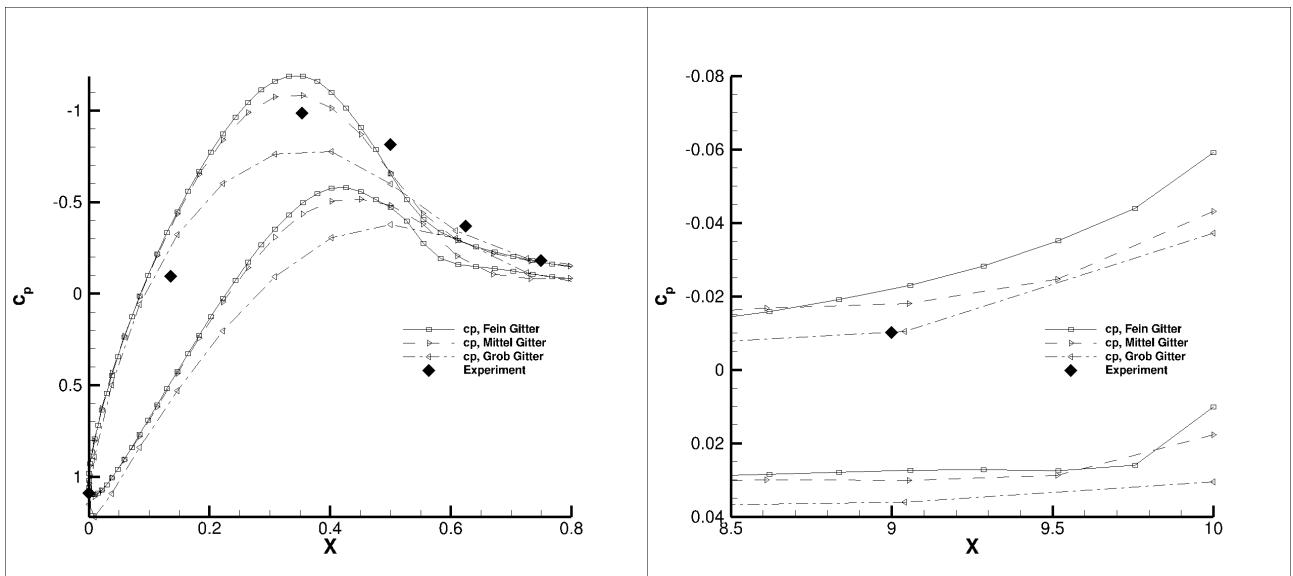


Abb. A.17: Drückbeiwert c_p Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, $AoA = 10^\circ$

Abb. A.18: Drückbeiwert c_p Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, $AoA = 10^\circ$

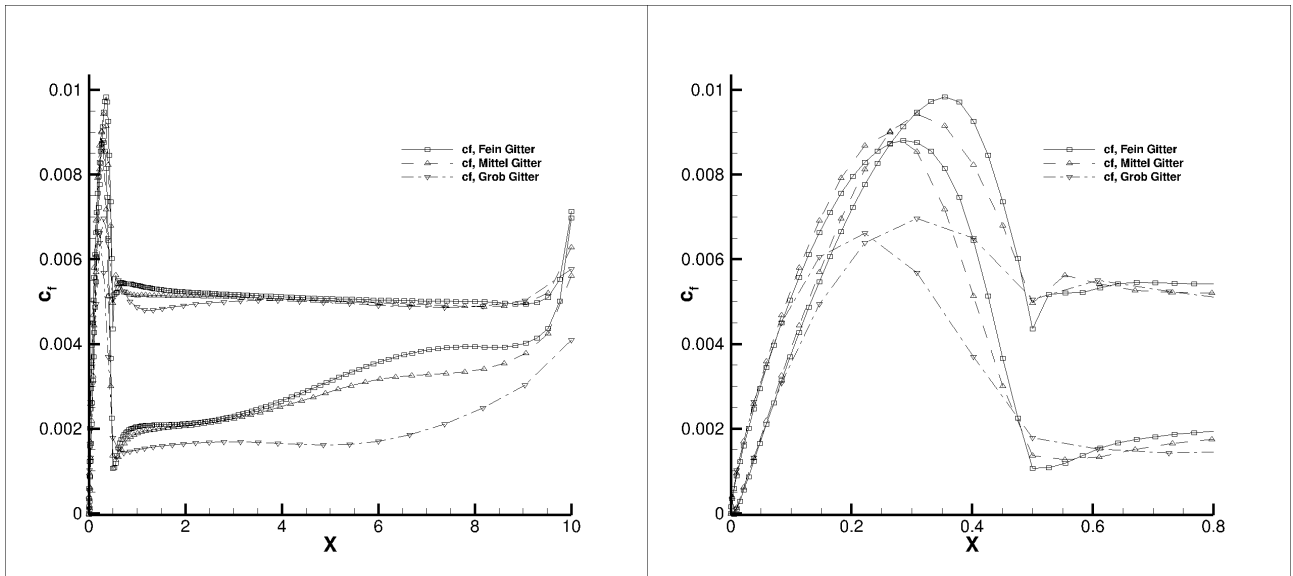


Abb. A.19: Reibungsbeiwert c_f Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, AoA = 10°

Abb. A.20: Reibungsbeiwert c_f Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, AoA = 10°

Konvergenzhistorie und konvergierte Beiwerten bei Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$, $AoA = 15, 0^\circ$

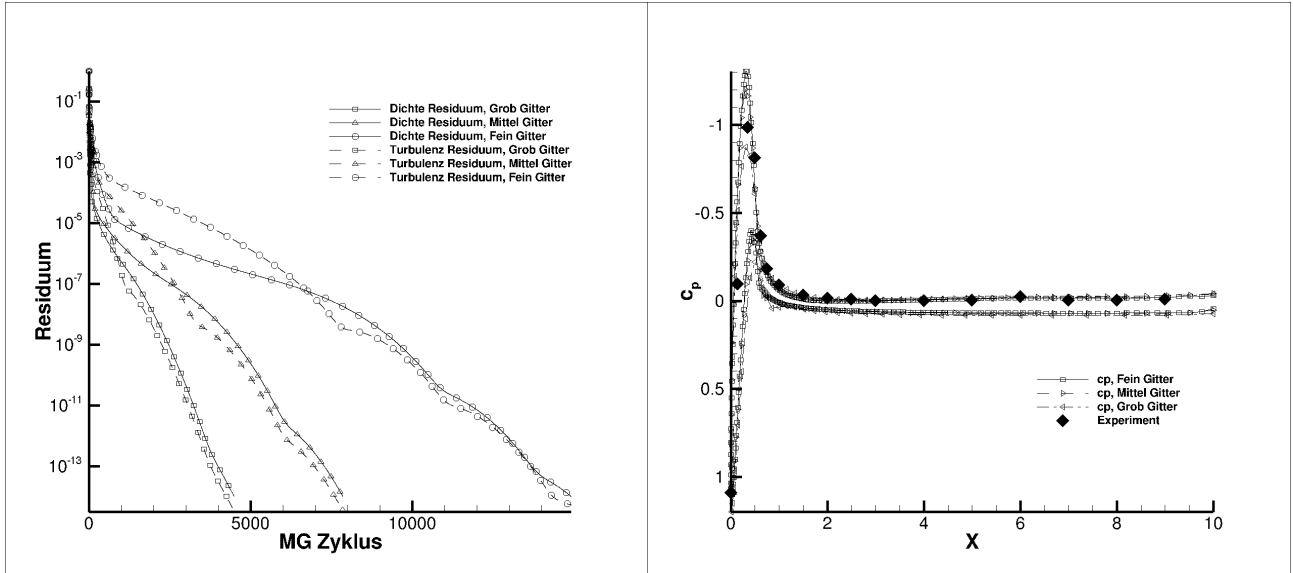


Abb. A.21: Konvergenzhistorie von Dichte und Turbulenz in Vergleich zu Gitterstufe, $AoA = 15^\circ$

Abb. A.22: Druckbeiwert c_p Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, $AoA = 15^\circ$

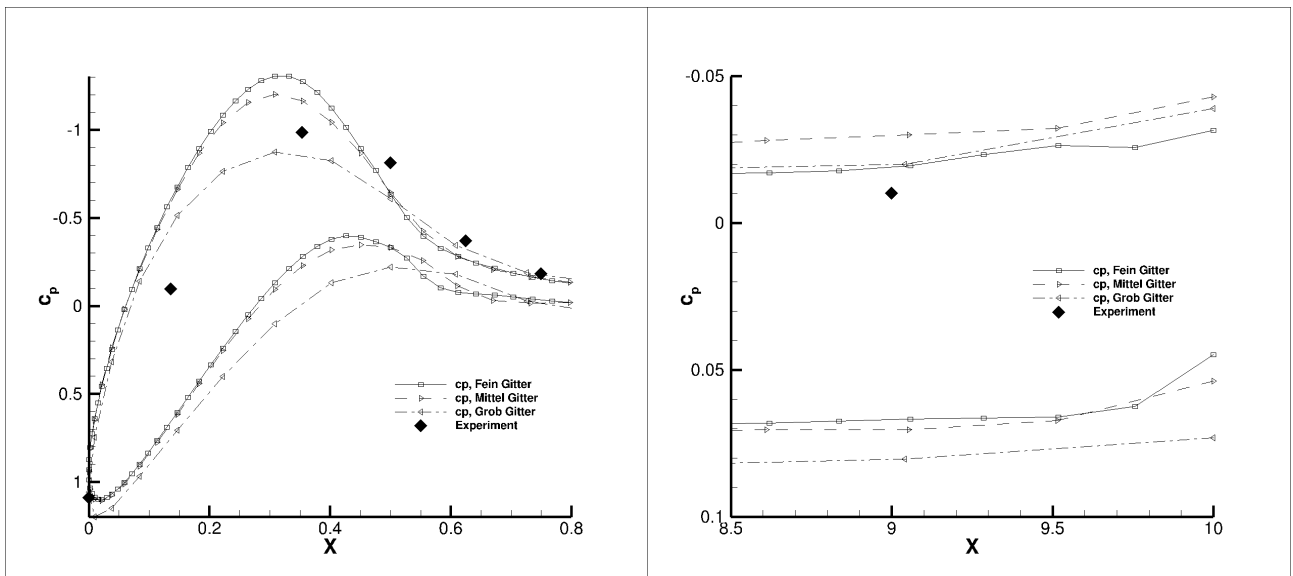


Abb. A.23: Druckbeiwert c_p Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, $AoA = 15^\circ$

Abb. A.24: Druckbeiwert c_p Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, $AoA = 15^\circ$

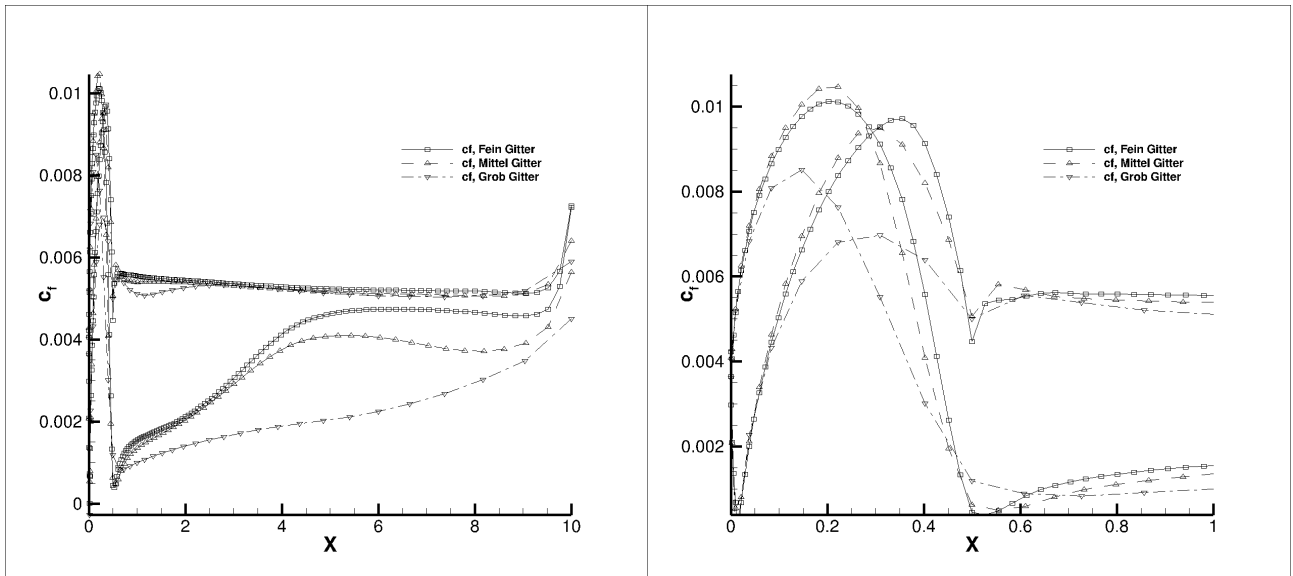


Abb. A.25: Reibungsbeiwert c_f Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, AoA = 15°

Abb. A.26: Reibungsbeiwert c_f Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, AoA = 15°

Konvergenzhistorie und konvergierte Beiwerten bei Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$, AoA = $19,0^\circ$

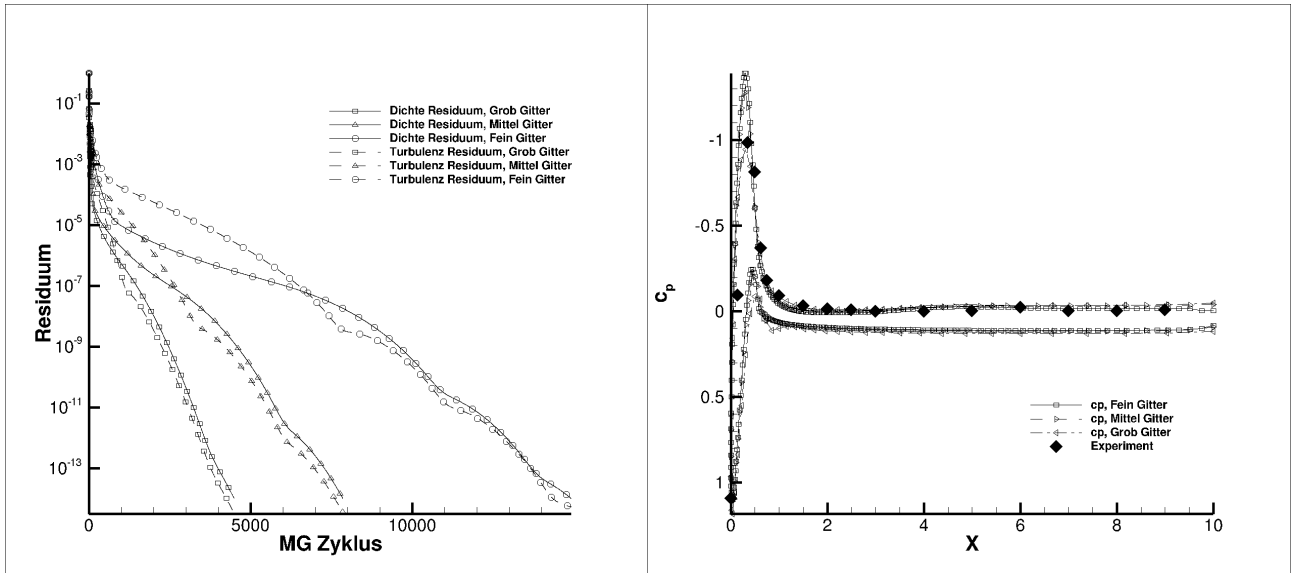


Abb. A.27: Konvergenzhistorie von Dichte und Turbulenz in Vergleich zu Gitterstufe, AoA = 19°

Abb. A.28: Druckbeiwert c_p Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, AoA = 19°

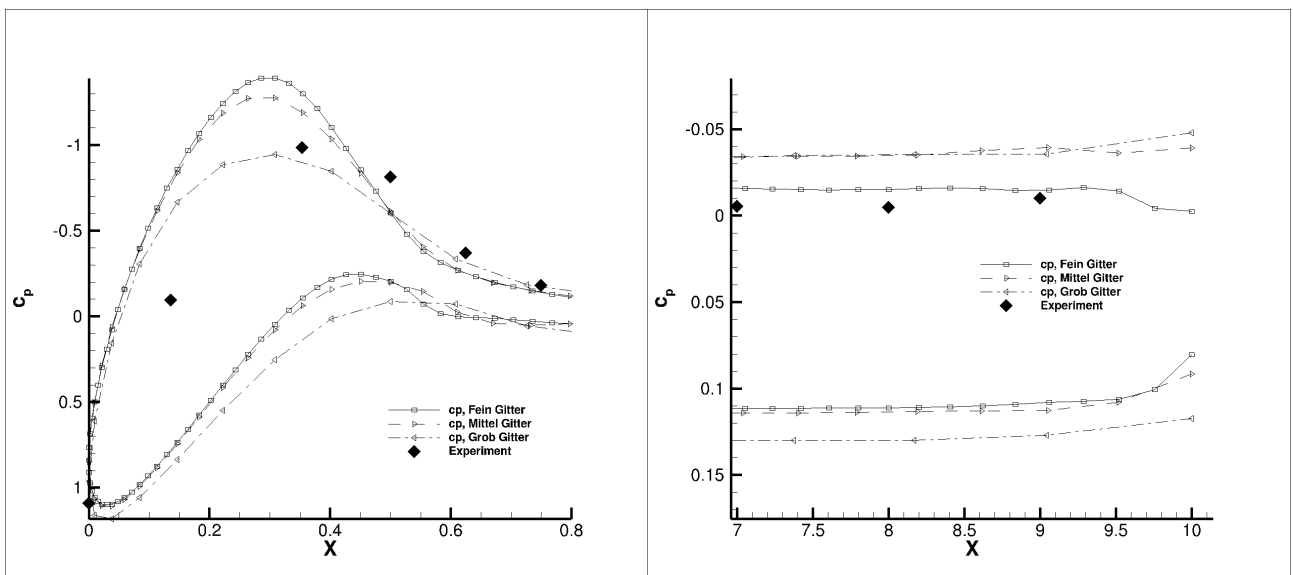


Abb. A.29: Druckbeiwert c_p Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, AoA = 19°

Abb. A.30: Druckbeiwert c_p Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, AoA = 19°

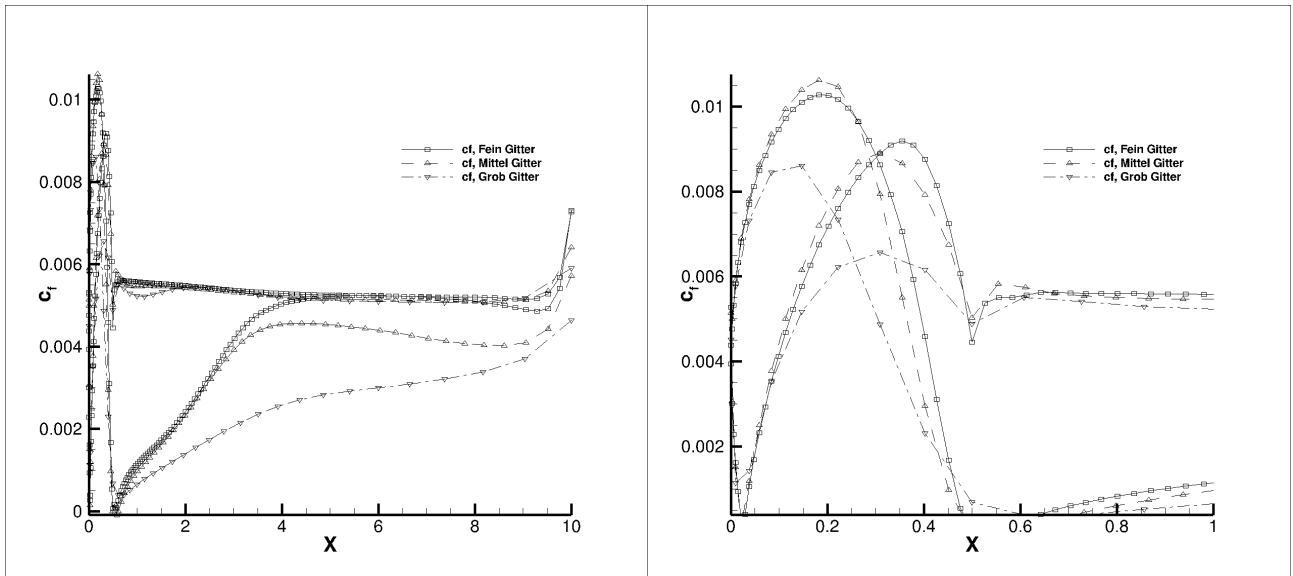


Abb. A.31: Reibungsbeiwert c_f Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, AoA = 19°

Abb. A.32: Reibungsbeiwert c_f Verteilungen auf dem Hemispher-Zylinder $\phi = 0^\circ, 180^\circ$ in Vergleich zu Gitterstufe, AoA = 19°

Geschwindigkeitsprofil in der Grenzschicht auf dem Hemispher-Zylinder x-, y- und z-Komponent u, v, w, $\text{AoA} = 0,0^\circ$

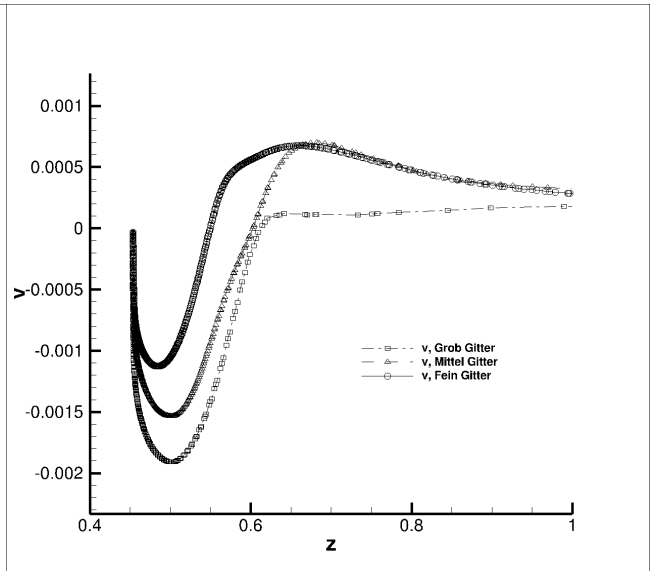
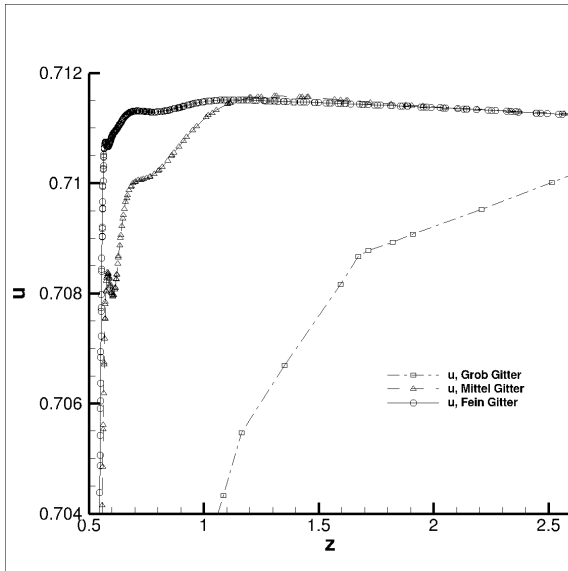


Abb. A.33: Geschwindigkeitsprofil von u auf dem Hemispher-Zylinder entlang z-Richtung am $x = 5, y = 0.21, 0^\circ$

Abb. A.34: Geschwindigkeitsprofil von v auf dem Hemispher-Zylinder entlang z-Richtung am $x = 5, y = 0.21, 0^\circ$

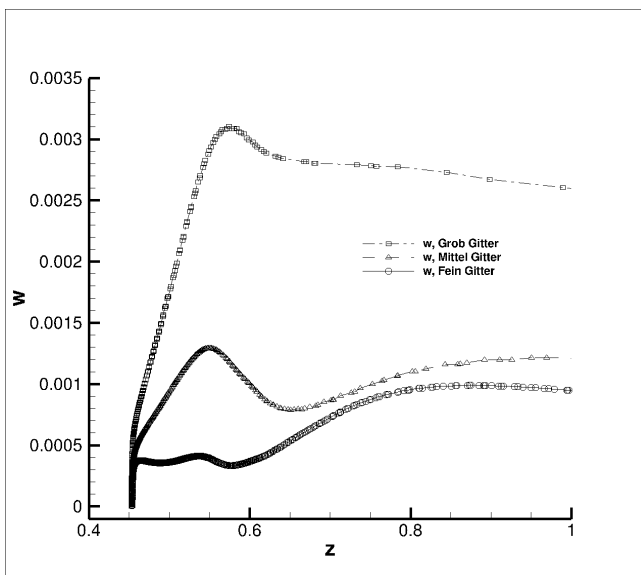


Abb. A.35: Geschwindigkeitsprofil von w auf dem Hemispher-Zylinder entlang z-Richtung am $x = 5, y = 0.21, 0^\circ$

**Geschwindigkeitsprofil in der Grenzschicht auf dem Hemispher-Zylinder x-, y- und z-Komponent
u, v, w, $\text{AoA} = 5,0^\circ$**

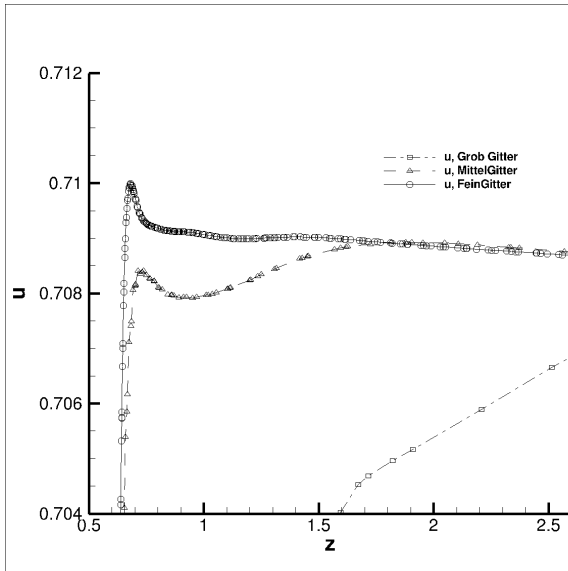


Abb. A.36: Geschwindigkeitsprofil von u auf dem Hemispher-Zylinder entlang z -Richtung am $x = 5$, $y = 0.21$, 5°

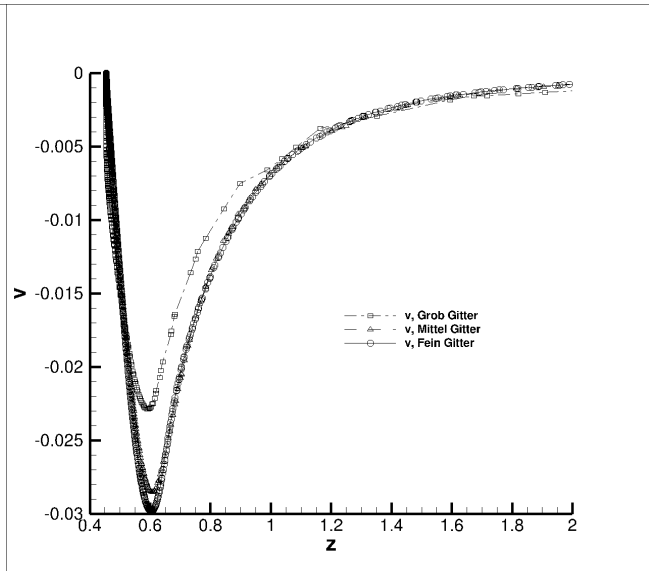


Abb. A.37: Geschwindigkeitsprofil von v auf dem Hemispher-Zylinder entlang z -Richtung am $x = 5$, $y = 0.21$, 5°

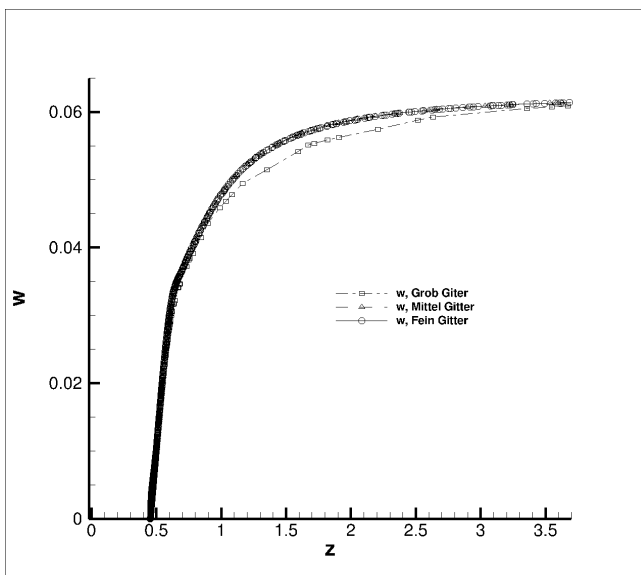


Abb. A.38: Geschwindigkeitsprofil von w auf dem Hemispher-Zylinder entlang z -Richtung am $x = 5$, $y = 0.21$, 5°

Geschwindigkeitsprofil in der Grenzschicht auf dem Hemispher-Zylinder x-, y- und z-Komponent u, v, w, $\text{AoA} = 19,0^\circ$

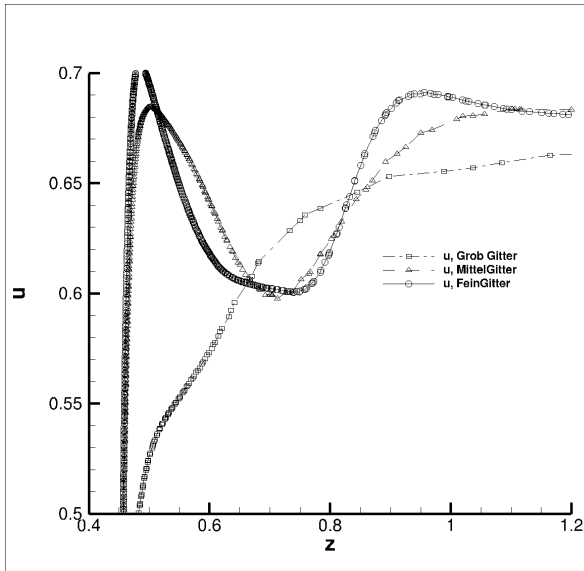


Abb. A.39: Geschwindigkeitsprofil u auf dem Hemispher-Zylinder entlang z-Richtung am $x = 5$, $y = 0.21$, 19°

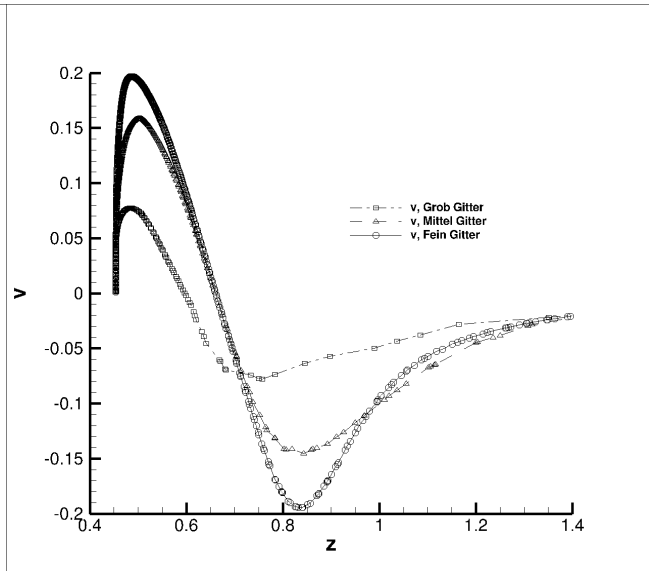


Abb. A.40: Geschwindigkeitsprofil von v auf dem Hemispher-Zylinder entlang z-Richtung am $x = 5$, $y = 0.21$, 19°

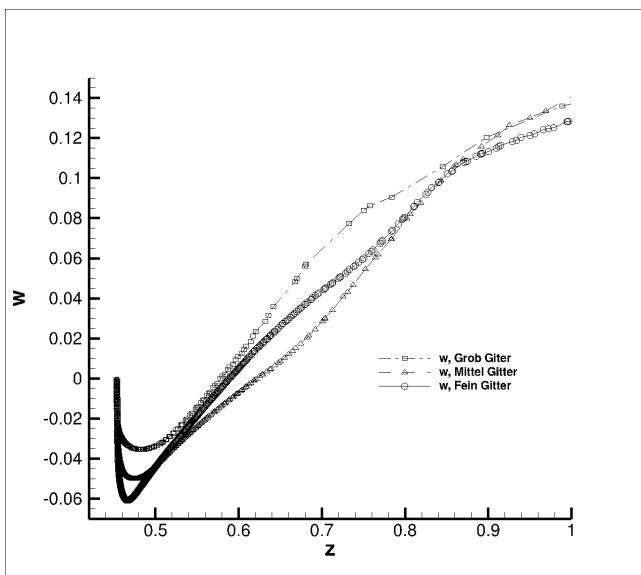


Abb. A.41: Geschwindigkeitsprofil von w auf dem Hemispher-Zylinder entlang z-Richtung am $x = 5$, $y = 0.21$, 19°